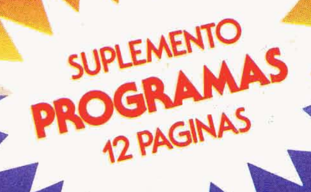
commodore Magazine

AÑO IV Núm. 39
Mayo 1987 300 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

Entropía e información

Margarita y el resolutor lógico



I.A. Máquinas inteligentes



TABLETAS GRAFICAS

ELCO calculadoras para estudiantes:

Por el precio de una calculadora sencilla



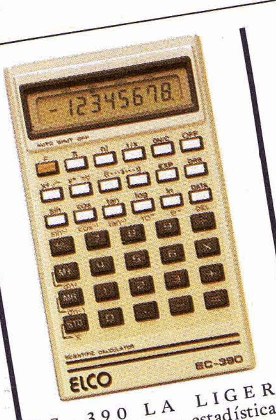
Garantía: UN AÑO.

2.990 ptas.

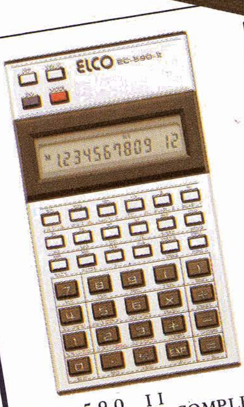
E C - 1 0 0 P N
LA CIENTIFICA ECONOMICA

Pantalla en LCD con 8 dígitos (5+2).
Funciones trigonométricas, logarítmicas, exponenciales y sus inversas. Grados centígrados, sexagesimales y radianes.
Factoriales, radicales, funciones estadísticas (media, varianza desviación típica).
AOS (sistema operativo Algebráico).
Apagado automático.
Alimentación con dos pilas normales. Duración aproximadamente 1 año.

31 FUNCIONES Y ESTADISTICA



E C - 3 9 0 L A L I G E R A
31 Funciones con estadísticas y 8 dígitos.
Apagado automático.
3.290 ptas.



E C - 5 9 0 I I
LA CIENTIFICA COMPLEJA
94 funciones y 12 dígitos.
Memoria constante.
Conversiones y cálculos en binario, hexadécimal, octal y décimal.
4.590 ptas.



E C S - 9 9 0 I I
LA SOLAR
94 funciones y 12 dígitos.
Conversiones y cálculo en binario, hexadécimal, octal y décimal.
Celdas solares de alta resolución.
5.590 ptas.



E C P - 3 . 9 0 0
LA PROGRAMABLE
Admite dos programas y 45 pasos de programación en memoria constante.
Con toma de decisiones.
64 funciones científicas y 10 dígitos.
6.590 ptas.

ALVARO SOBRINO



c/ Rufino González, 6
Telfs.: 204 76 56 y 204 05 70 - Telex 42489 ELCO E
28037 MADRID

Director:
Rubén Sanz
Colaboradores:
José D. Arias
Alejandro de Mora-Losana
Paloma Saco
Diseño:
Benito Gil
Montaje:
Angel Martínez
Edita
PUBLINFORMATICA
Bravo Murillo, 377 - 5.° A
Telf.: 733 74 13. Madrid - 28020
Presidente:
Fernando Bolín
Director Editorial
Revistas Usuarios:
Juan Arencibia
Director de ventas:
Antonio González
Jefe de Producción:
Miguel Onieva
Servicio al cliente:
Julia González - Telf.: 733 79 69
Coordinadora Publicidad:
Silvia Bolín
Publicidad Madrid:
Emilio García
Publicidad Barcelona:
María del Carmen Ríos
Pelayo, 12
Telf.: (93) 301 47 00
ext. 27-28 y (93) 318 02 89
08001 BARCELONA
Depósito Legal: M-6622-1984
Dirección, Redacción y Publicidad:
Bravo Murillo, 377 - 5.° A
Teléf.: 733 74 13
Distribuye: S.G.E.L.
Avda. Valdeparra, s/n.
Alcobendas. Madrid
Distribuidor en Venezuela:
SIPAM, S. A.
Avda. República Dominicana
Edif. FELTREE
Boleita Sur Caracas (Venezuela)
Distribuidora en Argentina:
Distribuidora Intercontinental
Sta. Magdalena, n.° 541
Buenos Aires (Argentina)

Fotocomposición: Consulgraf
Nicolás Morales, 34. 28019 Madrid
Fotomecánica: Karmat
Pantoja, 10. Madrid
Imprime: G. Velasco, S. A.
Esta publicación es miembro
de la Asociación de Revistas
de la Información, asociada
a la Federación Internacional
de Prensa Periódica FIPP.
El P.V.P. para Ceuta, Melilla y
Canarias, incluido servicio
aéreo, es de 300 ptas. sin IVA

Rogamos dirijan toda la correspondencia relacionada con suscripciones a Commodore Magazine. Edisa. Tel.: 415 97 12. López de Hoyos, 141-5. 28002 Madrid. Para todos los pagos reseñar solamente Commodore Magazine.
Para la compra de ejemplares atrasados dirigirse a la propia editorial Commodore Magazine, Bravo Murillo, 377 5° A. Tel.: 733 74 13. 28020 Madrid.

Editorial

Un año más se celebró en París el SICOB (Salón Internacional de Informática, Telemática, Organización de oficinas y Ofimática), desde el 6 al 11 de abril.

Entre las novedades que presentó Commodore, hay que destacar el AMIGA 500 y el AMIGA 2000. Dos nuevos ordenadores con un Motorola 68000 que aceptan tarjetas de expansión con procesadores 8088 y 80286, compatibles PC, XT Y AT respectivamente. Con la posibilidad multitarea característica de la serie AMIGA.

Esperamos poder ofreceros muy pronto las características particulares de estos nuevos ordenadores que intentan adaptarse a las costumbres de los usuarios PC y mostrar a la vez una magnífica resolución gráfica y una calidad de sonidos propia de los ordenadores AMIGA.

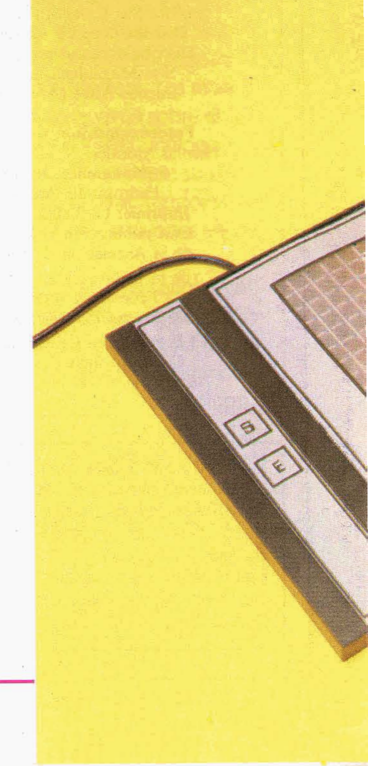
Con este número podrá realizar un acercamiento al periférico que más posibilidades de futuro tiene en este momento, las tabletas gráficas. Y que con mucha seguridad va a desplazar al Raton-mouse, que en muy poco tiempo ha adquirido una gran popularidad entre los usuarios que se acercan al mundo del software informático.



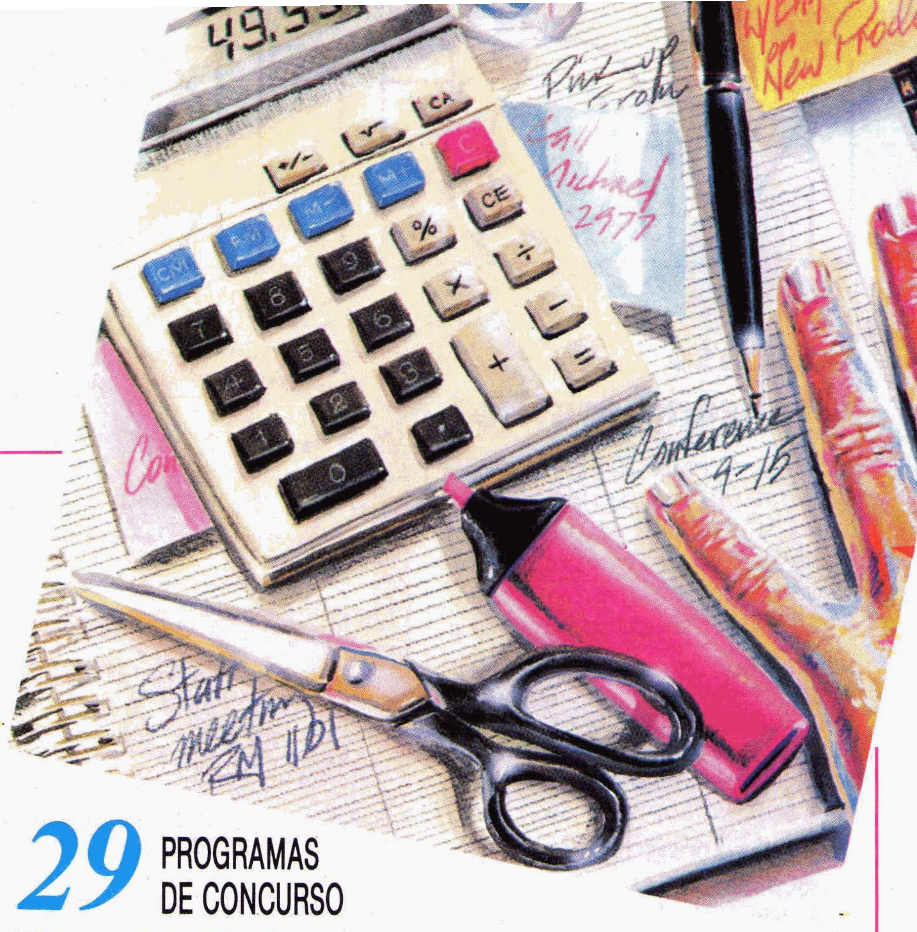
Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

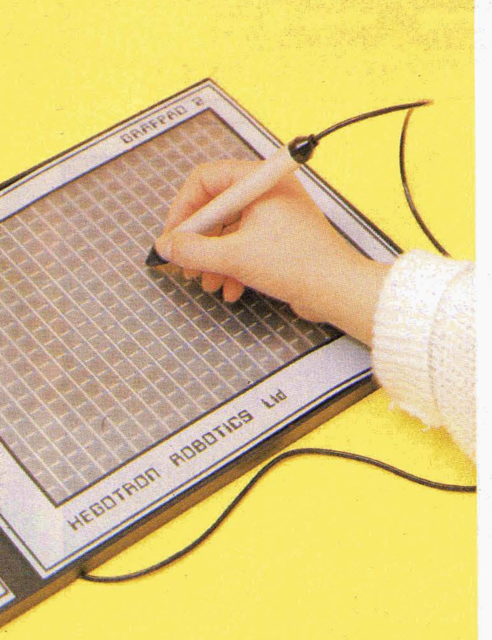
Suma






rio




GRAFPAD 2
HEGOTRON ROBOTICS Ltd

NOTICIAS

Concurso de diseño gráfico

La emisora catalana RADIO 4 de Radio Nacional de España ha convocado a través de su programa «L'ALTRA RADIO» espacio que tgrata temas de radioafición, DXsismo, Informática, comunicaciones y tecnología, un original concurso basado en el diseño por ordenador del logotipo de la emisora.

Con tal motivo, RADIO 4 y D.S.E.,S.A. han llegado a un acuerdo para la entrega de premios a los ganadores de entre los participantes.

D.S.E., S.A. patrocina la entrega de premios, otorgando:

- **2 ORDENADORES CON DICO DURO «BONDWELL BW-36»**
- **4 ORDENADORES CON DOS DISQUTS «BONDWELL BW-34»**
- **4 ORDENADORES PORTATILES CON DOS DISQUETS «BONDWELL BW-12**
- **2 IMPRESORAS «NEWPRINT I»**

Todo ello por un valor aproximado de **UN MILLON Y MEDIO DE PESETAS.**

Los que deseen tomar parte en este concurso, deberán elaborar un programa informático que dé como resultado la imagen que representa el logotipo de RADIO 4, a la que hay que dar animación y poner concurso, el logotipo y la partitura de la sintonía de la emisora como modelo, pueden solicitarse en:



RADIO NACIONAL DE ESPAÑA EN CATALUÑA.RADIO 4

Paseo de Gracia, 1
08007 Barcelona

Los trabajos deberán ser presentados en formato digital de ordenador, grabados en soporte magnético, con las categorías «HOME COMPUTER», «PC COMPATIBLES» o similares, a la dirección indicada de RADIO 4. El plazo de entrega de trabajos (un máximo de cinco por concursante) finaliza el día 15 de Junio de 1.987.



ATARI se establece en España

Después de unos pobres resultados de la mano de Investrónica, Atari decide iniciar su andadura europea como compañía propia.

Efectivamente, Ordenadores Ataria, S.A., se ha establecido en España como filial al 100% de Atari Corporation de EE.UU.

Comercializará la gama completa de sus productos en nuestro país garantizando, así, un soporte



y servicio adecuado a los usuarios de los ordenadores Atari. También estudia la posibilidad de establecer aquí su factoría europea.

Atari, pionera en productos electronicos de consumo, fué adquirida en 1984 por el equipo de Jack Tramiel, padre del Commodore Computer. Posteriormente desarrolló la línea de ordenadores Atari ST. La rápida gestión de datos unida a los destacables resultados en el tratamiento de gráficos y velocidad de proceso, lo convierten en el enemigo a batir. Con el precio situado a medias entre el Commodore-Amiga y los MSX de la II generación, los ordenadores Atari ST, pueden convertirse en la estrella del año.

Press Release

Una calculadora que funciona con agua

Todo el mundo ha oído hablar de las calculadoras solares, pero ¿y de las calculadoras de agua?.

Aunque parezca increíble, la firma de Hong Kong, Swak International Electronic Company Limited, ha conseguido desarrollar una calculadora LCD, resistente al agua, pero que funciona precisamente gracias a este elemento.

Su manejo es muy sencillo: se sumerge la calculadora en agua, que entra en su interior a través de unos agujeritos especiales, puesto que el aparato en sí es resistente al agua. En su interior hay un generador diseñado especialmente, que obtiene energía de este agua, y que mantiene la calculadora en funcionamiento durante tres meses. Tras este período, se vuelve a sumergir la calculadora en agua y obtendrá energía para tres meses más.

Se han obtenido ya las patentes para esta nueva y revolucionaria calculadora en los EE.UU. y el Reino Unido, principales compradores de Swank International.

Un ordenador para los deportistas

Ahora, la alta tecnología pone al alcance de los entusiastas del deporte una nueva manera de medir el esfuerzo realizado.

El «Sports Computer» es un aparato ultramoderno, diseñado para las personas que quieren ponerse en forma. Se trata de un aparato muy sencillo, que mide el ritmo de velocidad, proporciona un nuevo aliciente al deporte y vigila la salud del usuario midiendo su pulso. Se han diseñado varias versiones del «Sports Computer», para varios deportes específicos, tales como bicicleta fija, remo, jogging y esquí, y puede adaptarse a cualquier tipo de actividad.

Hay varios modelos de «Sports Computer» en el mercado, poseyendo esencialmente las mismas características. El modelo básico sólo tiene dos teclas, la de funcionamiento y otra que, en contacto con el pulgar, mide el pulso del usuario. Pero también puede disponer de escala de tensión del 1 al 10, cronómetro de cuenta atrás, taquímetro y monitor del ritmo cardíaco.

Dayton puede fabricar una variedad de modelos con 2-10 teclas, distintas funciones y de 1 a 4 pantallas. Actualmente, la empresa produce muchos modelos distintos, y vende el «Sports Computer» en EE.UU., Canadá, Australia, Nueva Zelanda y países europeos como Holanda, Suiza, Alemania, Noruega y Dinamarca.

Hong Kong Trade Development Council.
Balmes 184, 08006 Barcelona.
Telf.: 217 66 54.

IDEALOGIC

IDEALOGIC reduce precios

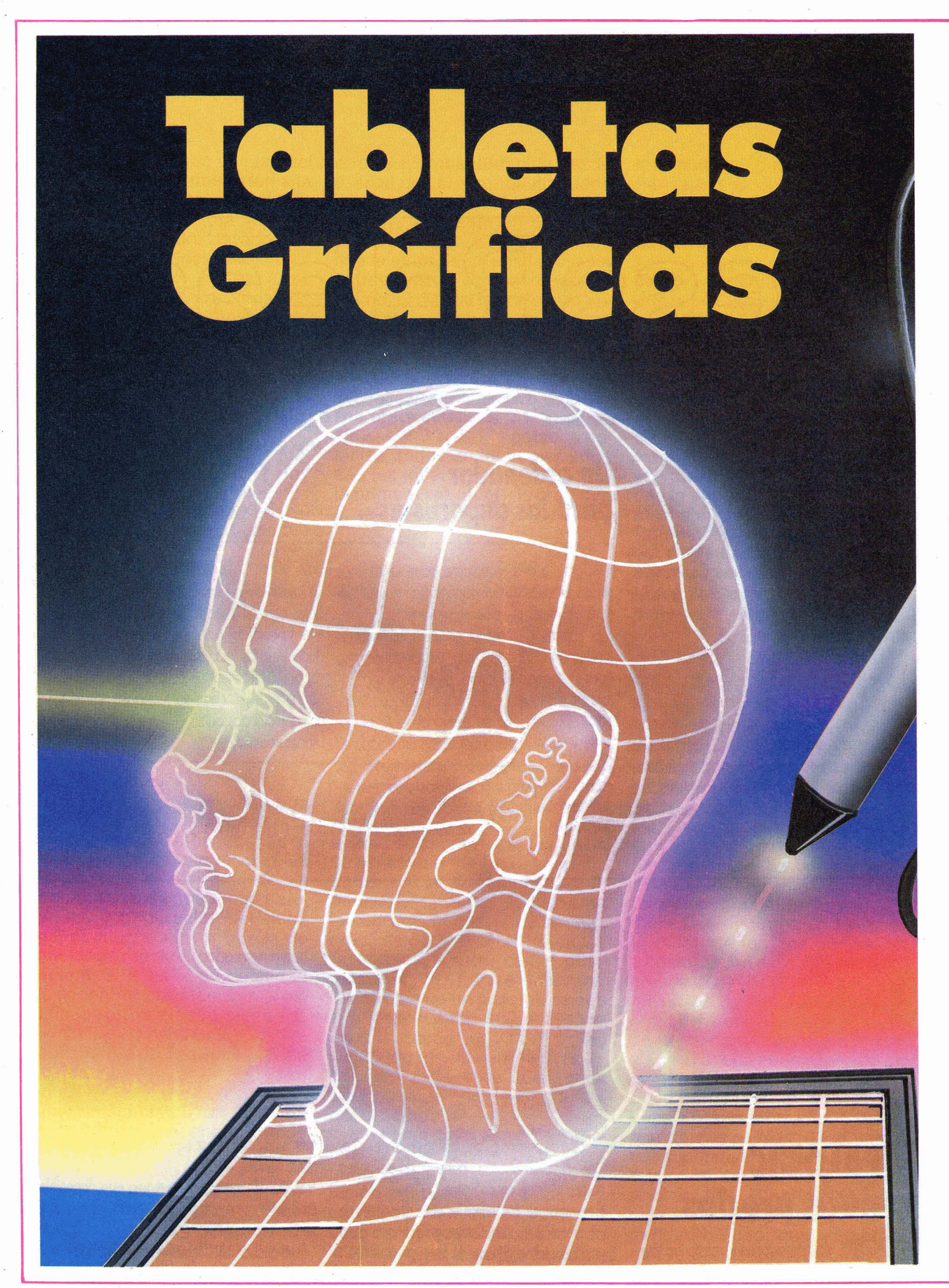
IDEALOGIC sigue apostando y apostando fuerte por el mercado doméstico con una importante reducción de precios en una buena parte de su catálogo. El que afecta a la parte lúcida del mismo y coloca el listón en 775 ptas.

El nuevo precio realmente les pone al alcance de un mayor número de usuarios y confiamos en la buena acogida por parte de los mismos.

El bajón es de por sí una importante noticia pero quizás hay otra que se hará oir todavía más, el paquete «SUPER 10».

SUPER 10 + Joystick de regalo está disponible en 3 sistemas: Amstrad, Commodore, Spectrum y en breve para MSX.

Y no acaban aquí las sorpresas, además han preparado los paquetes (2 + 1) y (3 + 1).


Tabletas
Gráficas

Fruto del esfuerzo por humanizar las relaciones entre el ordenador y el usuario, han visto la luz multidud de dispositivos creados para facilitar la entrada de datos en la memoria de un ordenador.
Sin lugar a dudas la tabla digitalizadora, también llamada tableta gráfica, es la reina de este tipo de periféricos y el complemento ideal del teclado de un ordenador.

Las tabletas digitalizadoras son unos dispositivos cuyo último fin, al igual que los lápices ópticos, ratones, *joysticks*, etc., es posicionar un cursor sobre una zona determinada de la pantalla.

Físicamente adoptan la forma de una tabla del tamaño de un folio, aunque las hay más pequeñas, solidaria a esta encontramos un elemento cursor que sirve para apuntar a una zona determinada de la tableta. El cursor suele adoptar la forma de un lápiz, aunque existen otros tipos, siendo particularmente útiles unos con una forma plana, parecidos a un ratón y que disponen de una zona transparente, similar a una lupa, con una cruz serigrafiada que indica el punto en el que se encuentra posicionado el cursor.

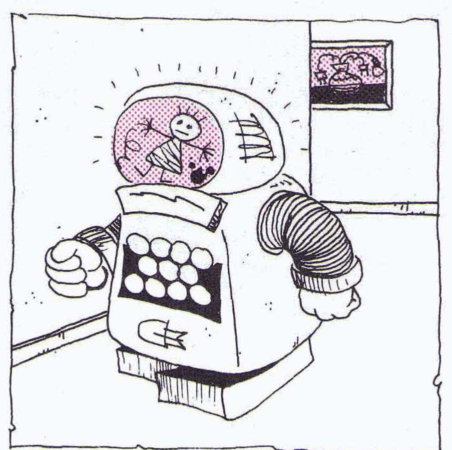
Es habitual encontrar uno o varios pulsadores bien asociados al elemento cursor o bien situados en los márgenes de la superficie activa. Todos estos elementos disponen a su vez del cableado que une la tabla al ordenador. Existen modelos con los interfases de comunicaciones standard; serie y paralelo, o el interfase específico para un ordenador en concreto como puede ser el CBM-64 ó CBM-128, el elemento cursor también suele ir unido a la tabla mediante un cable.

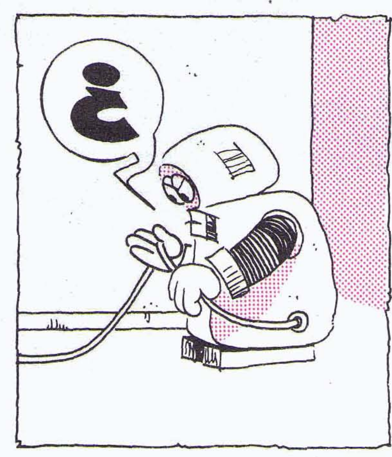
La utilización de una tabla digitalizadora es muy simple: cualquier actuación sobre la superficie activa de la tabla, mediante el elemento cursor, se ve acompañada de una acción en la pantalla del ordenador.

Existen dos grupos bien diferenciados entre estos dispositivos: por una parte las tabletas de presión como por ejemplo el Koala Pad y otras, en las cuales se determina la posición del cursor mediante una ligera presión sobre la tabla. Este tipo es particularmente útil, dado que se puede actuar sobre la tableta sencillamente con un dedo. Por otra parte encontramos las tablas que se sirven de medios electrónicos para determinar la posición del cursor bien por ultrasonidos, cargas electroestáticas o por ondas electromagnéticas, en las que apoyando sobre la superficie de la tableta el elemento cursor, podemos posicionarnos en cualquier lugar de la pantalla del ordenador, pertenece a este grupo el GRAFPAD II.

La precisión de estas últimas es mayor, encontrando este principio de actuación en la mayoría de tabletas profesionales.

Es lógico pensar que una sencillez de manejo tal, junto con una gran precisión a la hora de posicionar el cursor en la panta-







lla, el campo en el que se obtiene mayor rendimiento a las tablas digitalizadoras es el de proceso de gráficos por ordenador.

Para el caso concreto del CBM-64 ó CBM-128, el *software* que acompaña habitualmente a las tablas digitalizadoras suele ser, como apuntábamos anteriormente, *software* de proceso de gráficos.

La composición de este *software* está dividida en dos partes, por un lado encontramos un menú con las típicas opciones de este tipo de programas: líárea de dibujo que en algunos paquetes está formada por dos pantalla gráficas, siendo posible tener dos dibujos diferentes en la memoria del ordenador y copiar zonas de un dibujo a otro. neas, círculos, arcos, elipses, rectángulos, junto con las opciones de modificar los colores, rellenar superficies y las habituales de edición como trasladar zonas del dibujo, copiarlas, borrarlas, efectos de simetría, etc. y algunos permiten efectos más sofisticados como *sprites* diferentes fuentes de texto de diversos tamaños, etc.

Habitualmente las opciones del menú están representadas por un pequeño gráfico —icono— que nos indica la acción que realiza (una brocha para rellenar, un pincel para dibujar, etc.). La selección se lleva a cabo posicionando el cursor sobre la opción elegida y actuando sobre algún pulsador de la tabla o del teclado.

Por otro lado encontramos pantallas gráficas, siendo posible tener dos dibujos diferentes y copiar de un dibujo a otro.



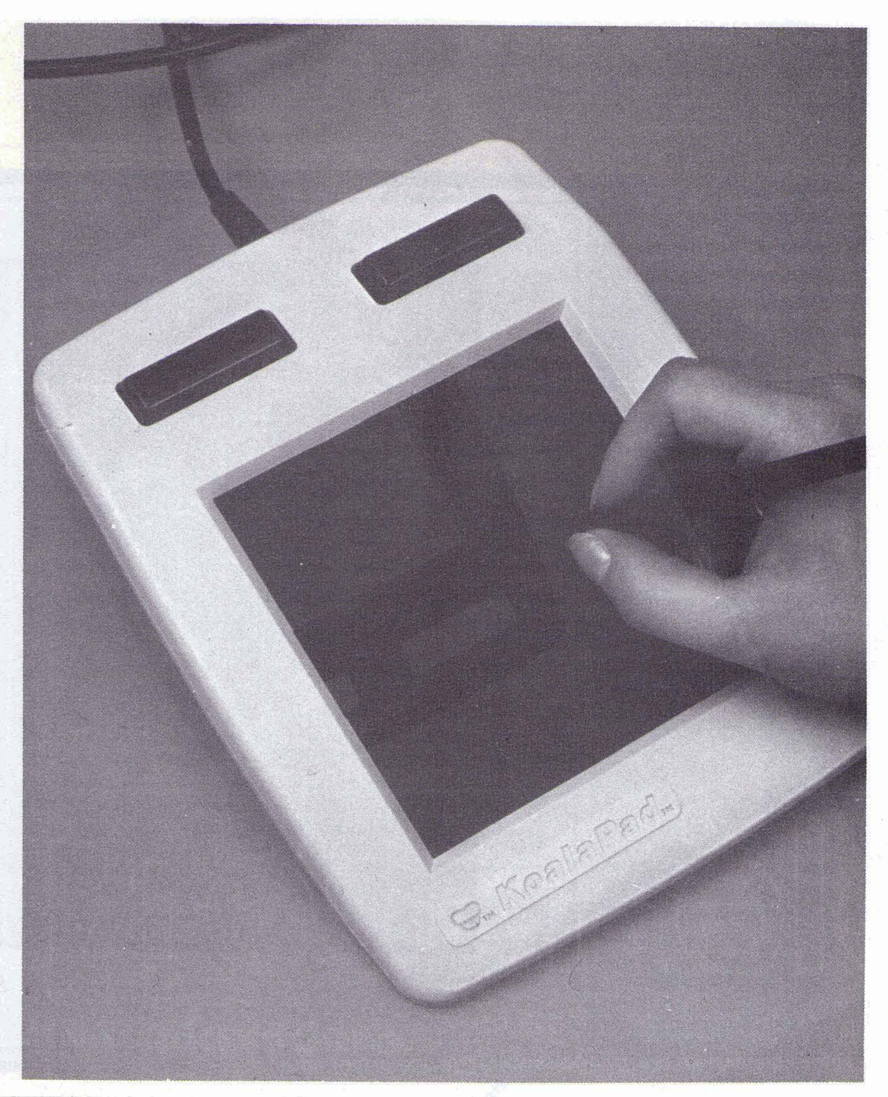
Se trata de la pionera dentro de las tabletas digitalizadoras para CBM-64. Es del tipo de presión, siendo la superficie sensible algo menor del tamaño de una cuartilla.

Su reducido tamaño le permite acoplarse con facilidad a nuestra mesa de trabajo.

Hardware:

La tabla se conecta al ordenador en un *port* de juegos, mediante el conector apropiado, la tabla dispone de dos pulsadores y el elemento cursor no está unido físicamente a la tabla, por lo tanto con cualquier objeto se puede actuar sobre ésta, incluso con un dedo.

La superficie activa está compuesta por una serie de pistas conductoras dispuestas en paralelo con los dos ejes, el hori-



zontal y el vertical, formando una especie de rejilla. En estado de reposo no existe contacto físico entre ninguna de las pistas, pero al ejercer una ligera presión una o varias de las pistas verticales —tocan— las pistas horizontales. Este principio de actuación es similar al de un pulsador, y la parte electrónica de la tabla es la encargada de trasladar las coordenadas donde se produjo el contacto a las coordenadas reales de la pantalla.

Software:

Además del programa de diseño gráfico que acompaña a la tableta, en forma de disco en la primera versión, han ido apareciendo otros programas de diverso tipo que soportan la utilización de la tableta. Pese al paso inexorable del tiempo aún tenemos en el mercado este eficaz periférico, que continúa siendo en relación calidad-precio una opción a tener en cuenta.



Esta tableta de origen inglés, es la segunda versión del mismo dispositivo que apareció hace dos o tres años en Inglaterra. Existen modelos para varios ordenadores, incluso PC, y es un standard dentro de las tabletas gráficas de bajo precio.

Hardware:

En la parte izquierda, encotramos dos pulsadores y el elemento cursor se halla unido a la tableta mediante un cable.

Es del tipo de detección electromagnética, siendo la superficie activa del tamaño aproximado de un folio.

De nuevo encontramos un entramado de conductores formando una rejilla, pero esta vez no existe contacto físico entre unos y otros, ya que el principio de detección es diferente.

La superficie activa se explora en intervalos de tiempo regulares, primero se exploran todas las filas y a continuación todas las columnas. La exploración así realizada a gran velocidad mediante una señal de onda cuadrada, es capaz de detectar la presencia del elemento cursor, por una deformación de la forma de la onda exploradora, y midiendo el tiempo transcurrido desde que se detecta su presencia hasta que se han barrido todas las filas-columna, se puede determinar el punto en el que está posicionado. De nuevo la parte electrónica es la encargada de trasladar estas coordenadas a las reales.



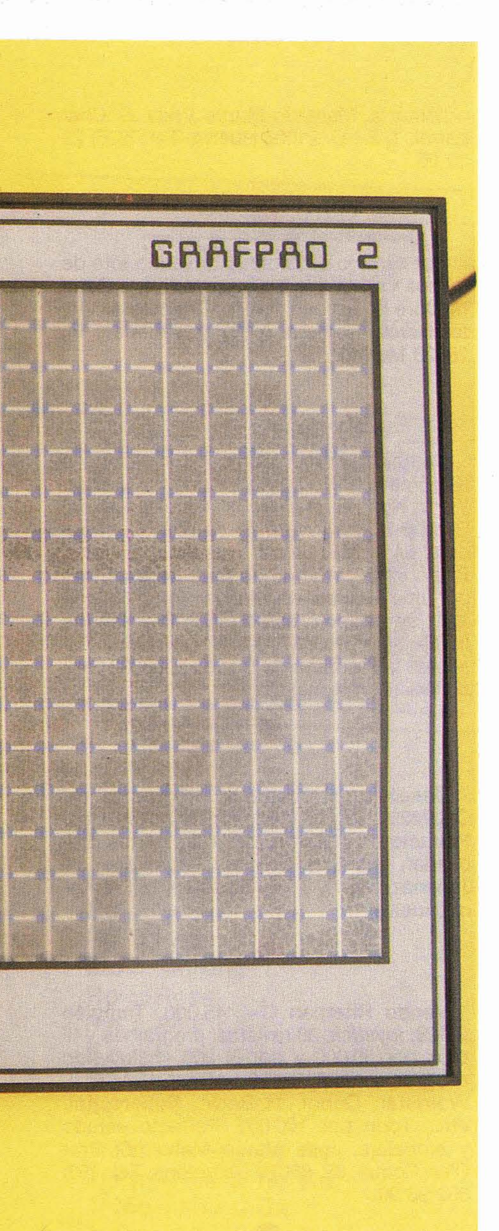
La parte electrónica está compuesta por un circuito integrado de diseño propio, que sustituye a la gran cantidad de circuitería que sería necesaria para realizar la misma función y varios componentes más en el cursor en forma de lapicero. Se conecta al ordenador por el port de usuario.

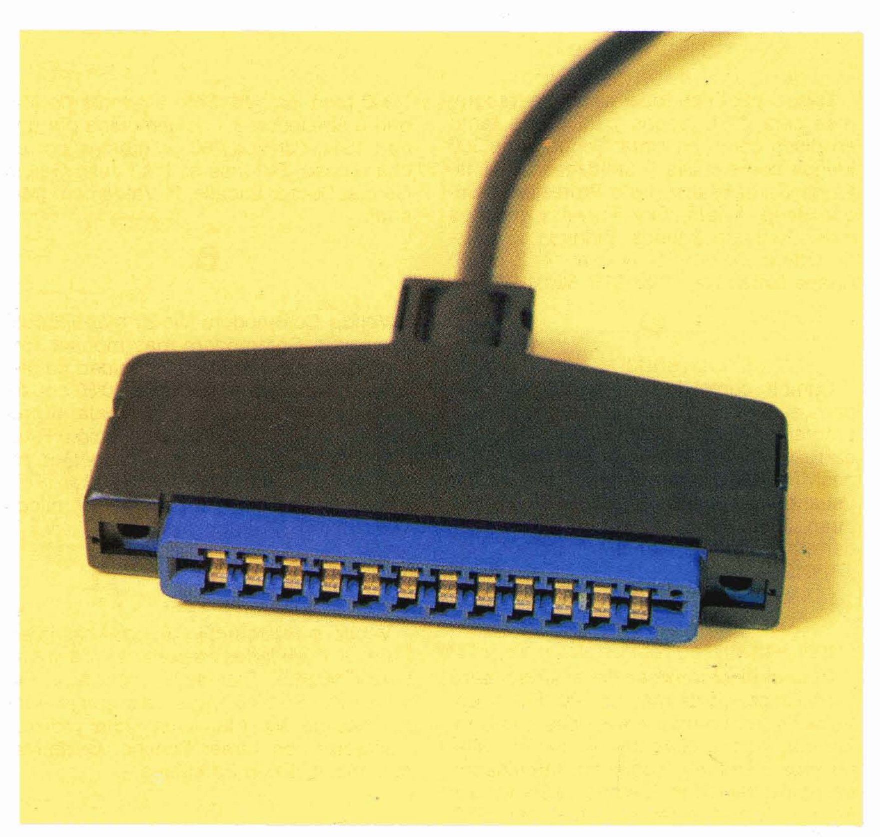
Existe la posibilidad de poner sobre la superficie activa plantillas serigrafiadas con las opciones del *software* que se esté utilizando.

Software:

El *software* que acompaña a la tableta es uno de los mejores de este tipo.

El menú de selección en forma de iconos tiene multitud de opciones aparte de las standard en proceso de gráficos con cua-





renta opciones primarias, algunas de las cuales dan paso a otras opciones secundarias.

En conjunto es un paquete con unas características excepcionales, incluso apropiado para hacer algunos pinitos en el campo del CAD.

Enrique Fernández Rasero.
Técnico Especialista Aeronaútico.

¿te interesa?

Deseo cambiar toda clase de programas para C-64, juegos y utilidades tanto en disco como en cinta. Tengo unos 300 juegos comerciales y utilidades de la talla de Game Maker, Loto Profesional, Super Base, Koala, Sky Travel y muchos más. Contesto a todos. Dirigirse a: Agustín García Cordón. C/ Nueva, 14, 2. 50700 Caspe (Zaragoza). Tel.: (976) 63 01 98.

●

Cambio curso de introducción al Basic para el C-64 y una cinta o disco con 15 programas (Logo, One on One, etc.) por cartucho Simon's Basic con instrucciones. También quiero formar un club de usuarios gratuito en Alcorcón. Dirigirse a: Juan José Pulido Labradas. Plaza Ondarreta, 5, 2.° A. Alcorcón (Madrid). Tel.: (91) 619 16 54.

●

Deseo intercambiar programas para C-64. Dispongo de más de 300 títulos, entre ellos las últimas novedades como Infiltrator, Cobra, Gauntlet, Breakthru, Miami Vice... Prometo contestar. Interesados escribir a Santiago Meroño León. C. Los Fontes, Torre Pacheco (Murcia). Tel.: (968) 57 75 64.

●

Vendo por 30.000 ptas. un SVI 728 de 80 K Ram o cambio por un Commodore 64/128 K. Llamar a Alfonso Montoya Quiles. Tel.: (951) 43 16 12 o escribir a Plaza Nueva, 5. Albox (Almería).

●

Desearía cambiar juego de calidad como: Pitstop II, Uridium, Ghosts'n Goblins, Kung-Fu Master y otros. Dirigirse a: Jordi Garriz Melcan. Plaza Reus, 1, 5 1. Cornella (Barcelona).

●

Compro Commodore 64 en buen estado y funcionamiento. Estaría dispuesto a ofrecer entre 10.000 a 15.000 ptas. Además compraría Unidad 1541, por la cual pagaría de 15.000 a 18.000, sólo provincia Barcelona o alrededores. Interesados contactar con Carlos Felipe Spada. C/ Rosellón, 290. 08037 Barcelona. Tel.: (93) 257 75 51 (comidas y cenas).

●

Me urge mucho vender 15 *interfaces* copiadores para el Commodore 64. Confianza y efectividad al 100%. Los vendo a 3.000 ptas. unidad. Sólo a gentes de Madrid o alrededores. Los cambiaría por unidad 1541. Ofrezco 700 programas por dicha unidad. Dirigirse a: José Julio García García, Doctor Lacalle, 2. Valdemoro (Madrid).

●

Vendo Commodore Vic-20 más módulo expansión Commodore más monitor fósforo verde con sonido más unidad de disco + *datassette* + cartucho 40/80 col. de Ferret Moret más cartucho ayuda al programador más 32 Kb. ampliación RAM más código máquina más *cassettes,* revistas, libros, etc. Todo 50.000 ptas. Dirigirse a: José Carmona Saceta. C/ Amilcar, 146, 1 3. 08032 Barcelona.

●

Vendo o intercambio juegos originales. Ultimas novedades importadas de Inglaterra (Tobanck, Bismarch, Crusade in Europe, etc.). Sólo 400 ptas. cada juego. Tengo más de 300 títulos en cinta y disco. Contactar con Xavier Sancho. C/ Muntaner, 153, 4. 08036 Barcelona.

●

Vendo Commodore Plus/4 casi nuevo e incluso manuales de programación y funcionamiento de los programas integrados; Base de Datos, Tratamiento de Textos, Hoja electrónica y Gráficos. Perfecto estado por sólo 15.000 ptas. Interesados escribir a: Carlos Felipe Spada. C/ Rosellón, 290. 08037 Barcelona o llamar al tel.: (93) 257 75 51 (comidas y cenas).

●

Intercambio programas de juegos originales para el C-64, como por ejemplo: Boulder Dash, Uridium, Zorro, Super Zaxxon, etc., los cambio por Summer Games I-II, Fist II o Winter Games. También intercambio juego y utilidades en turbo con carga principal de Turbo V-3 o Turbo Tape, como por ejemplo: Broad Street, Breakdance, Combat Linx, etc. En total dispongo de 20 programas originales y 40 programas turbo. Todo ello también lo vendo por 20.000 ptas. Escribir a: Rodrigo Pérez Sánchez.
C/ Poeta Gerardo Diego, 1, 4.° A. 39011 Santander (Cantabria). Tel.: (942) 33 08 58.

●

Estoy muy interesado en intercambiar juegos para Commodore 64 en cinta. Poseo muchas últimas novedades. Prometo contestar las cartas. Interesados escribir o llamar a: Mauricio Martín Vega. C/ Chaparral, 1, 5.° C. 21006 Huelva. Tel.: (955) 23 19 00.

●

Por cambio de ordenador vendo lote de discos con utilidades y juegos para Commodore 64 a buen precio. Dirigirse a: Víctor García Romero. Avda. de América, 14. 28028 Madrid.

●

Cambio Summer Games II, A view to a Kill, Veam rider, Rambo, Kung-Fu Master, Yie ar Kung-Fu, Misión Imposible, Airwolf, Pitstop II, Sprectrum 48 K (para Commodore 64 y 128), Logo, Hes-Mon, Simon's Basic y 132 programas más de este tipo por una unidad de discos 1441, 1540 ó 1571 en perfecto estado. Los interesados llamar al tel.: (93) 870 61 59 o escribir a Paseo de la Montaña, 141, ático 2. 08032 Granollers (Barcelona).

●

Desearía intercambiar programas de C-64/128, poseo las últimas novedades. Prometo contestar a todas las cartas. Sebastián Mendoza García. C/ Juan Ramón Jiménez, blq. 1, 3 izq. Palos de la Frontera (Huelva).

●

Vendo Riteman C+ 45.000. También C-128, *joystick*, 30 revistas, programas y libros por 50.000 y por 65.000, incluyendo programas del C-128, como dBASE II, Wordstar, Cobol, MsBasic, Superscript, etc... Todo por 100.000. Perfecto estado y embalaje. José Marsá Mallol. C/ Prat D'en Roqué, 32. 08029 Barcelona. Tel.: (93) 352 98 90.

●

Desearía intercambiar programas e información con chicos que tengan un Commodore 16 K. Yo tengo programas, juegos y revistas del Commodore 16 K. Interesados escribir a Ignacio Lazo Montoya. Ctra. Cortes Venta el Chano. Jerez de la Frontera (Cádiz).

●

Por cambio de ordenador, vendo cable 40/80 columnas para el C-128 y de paso doy acceso a una interesante y selecta programoteca. Llamar al Tel.: (94) 469 49 74 y preguntar por José Ignacio Saiz.

Vendo los números 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 27, 31, 32, 34 de Commodore Magazine (100 pts./unidad), los números 11, 12, 15, 17, 21, 23, 25, 27, 30, 34 de Commodore World (150 pts./unidad) y los números 1, 11 y un extraordinario de verano de Input Commodore (125 pts./unidad). Llamar al teléfono: (91) 202 58 02. Preguntar por Paco o Eduardo.

Vendemos juegos desde 25 a 200 pts. de Commodore 64K y Dragon modelo 64K. Interesados llamar o escribir a Jorge Tostado. C/. Joaquín Crosal, 15-17-1.º-1.ª. 08950 Esplugues de Llobregat (Barcelona). Tel.: (93) 371 45 75.

Urge comprar unidad de discos. Puedo ofrecer un lote de programas valorado en 130.000 pts. a cambio de unidad de discos; o de impresora; o de monitor color; o de cualquier periférico para CBM 64. Cambio CBM-64 por CBM-128 pagando la diferencia y regalo programas. Vendo videopac Philips G-7400 convertible en ordenador, regalo programas. Intercambio *software* para CBM. Tengo todas o casi todas las últimas novedades. Francisco Javier Bernal Malillo. C/. Echegaray, 1. Cantalejo. Segovia. Tel.: (911) 52 04 04.

Vendo impresora Seikosha GP-100 VC especial Commodore, como nueva. Pocos meses de uso por sólo 25.000 ptas. Regalo además 300 juegos en cinta para C-64 (últimas novedades). Llamar mañanas de 8 a 3 al teléfono: (93) 302 32 00 ext. 280. Preguntar por Gabriel.

Deseo intercambiar programas y pokes para C-64 y C-128. Poseo unidad de disco. Escribid a: Pedro Moreno Espigares. C/. Macia, 66. 29013 Málaga.

Nuevo club de usuarios del Commodore 64, con una programoteca de más de 1.000 programas para intercambio de programas, trucos, cargadores, mapas, pokes, etc. Creación de una revista para los socios. Sin ninguna cuota, de programas, enviadnos vuestras listas. Prometemos contestaros facilitandoos información sobre el grupo. Dirigirse a: José. Apartado de correos, 195. Lérida.

Deseo comprar la colección completa de la Revista Commodore World y los 13 primeros números de Commodore Magazine. Mi dirección es: Lluís Privat. C/. Canemeres, 34. 17820 Banyoles (Gerona).

Vendo equipo Commodore 64, *joysticks, diskettes,* curso Basic en cintas y fascículos, revistas y juegos por cambio de ordenador. Llamar a Salvador de 10 a 12 mañanas al teléfono: (93) 353 86 25.

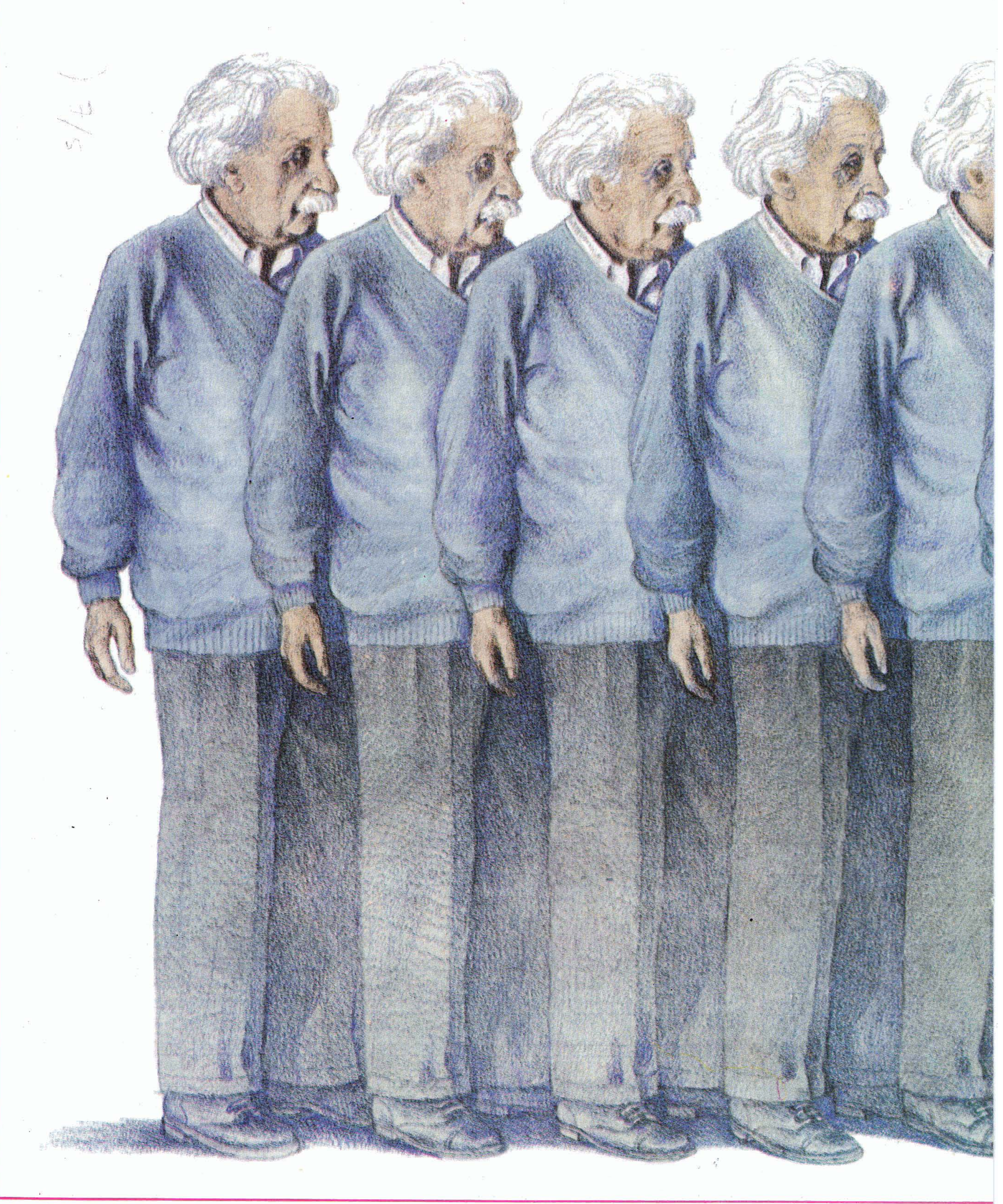
ANUNCIOS GRATUITOS

Todos los anuncios (compras, ventas, cambios o comunicaciones de clubs de usuarios, etc.) que van en esta sección, deben tener un máximo de cuarenta palabras. Con el fin de facilitar la transcripción de los anuncios hemos recuadrado cuarenta espacios para que en cada uno vaya una palabra. Después, recortar y mandar a:

commodore Magazine

ANUNCIOS GRATUITOS
C/ Bravo Murillo, 377, 5.° A
28020 MADRID

Primera parte





Máquinas inteligentes I.A.

En general, los mosquitos no son considerados como seres inteligentes. Sin embargo, hasta ahora, los más grandes y veloces ordenadores no han conseguido igualar la potencia de procesamiento de un mosquito, con su habilidad de volar, navegar, aterrizar, para identificar blancos, compañeros y áreas de reproducción, para sobrevivir como especie durante millones de generaciones en este mundo tan impredecible y cambiante. El mosquito apenas tiene unas pocas decenas de miles de neuronas cerebrales, el ser humano tiene cerca de un billón.

Los científicos abocados a la tarea de lograr máquinas capaces de pensar como los humanos se enfrentan a lo que tal vez sea la más ardua tarea que se haya fijado jamás el hombre. Virtualmente, cada investigador en el campo de la Inteligencia Artificial tiene sus propias ideas favoritas acerca del mejor camino a seguir para alcanzar esa meta. Algunos parecen convencidos de que los ordenadores inteligentes deben tener movilidad y ser capaces de computar la información sensorial. Otros argumentan que lo fun-



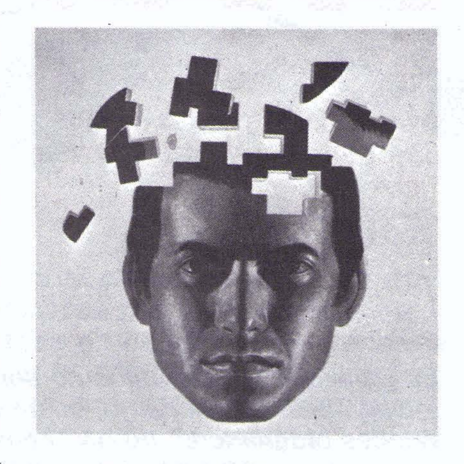
damentalmente importante es la lógica, mientras que abundan los que opinan que más que lógica, los ordenadores deben aprender los significados antes de adquirir una conducta inteligente.

Hans Moravec, investigador científico de la Universidad Carnegie-Mellon, es de los que creen que la senda hacia la IA pasa por la construcción de robots móviles, máquinas con autonomía de desplazamiento en el mundo real. Más optimista que la mayoría de sus colegas, Moravec predice que dentro de 20 años habrá robots con un nivel humano de inteligencia.

Ahora, en la mitad de su treintena, se puede decir que Moravec ha pasado las cuatro quintas partes de su vida dedicado a construir robots. El primero lo hizo cuando apenas tenía 7 años, pero el primero «en serio» (con cuerpo, motor, batería y cables) tres años más tarde.

A partir de entonces, no ha cesado de hacer robots, primero en la Universidad de Stanford y luego en la Carnegie-Mellon. Entre los éxitos de Moravec se encuentra Stanford Rover, el primer robot equipado de visión, con capacidad para percibir y reconocer obstáculos y replantear el recorrido para evitarlos. Para este especialista, la movilidad es un factor clave, pues fuerza la adaptabilidad general. Le parece imperdonable que se engañe al público calificando de «inteligentes» a algunos ordenadores, que en realidad no son capaces de llevar a cabo más de una tarea programada por vez. La movilidad supera a la percepción sensorial, las pautas de reconocimiento y el aprendizaje, «talentos» todos ellos que el ser humano comparte con animales inferiores aunque sean extremadamente difíciles para los autómatas. Por ejemplo, a un autómata le es mucho más difícil que a un individuo de inteligencia relativa el aprender a jugar o a diagnosticar una enfermedad.

Moravec cree que los investigadores en IA están innecesariamente desmoralizados por el descubrimiento de que algo que resulta natural para los hombres se resista a ser ingresado al código de un ordenador. Este era uno de los temas que en tiempos solía discutir con su consejero de tesis, **John McCarthy,** uno de los fundadores y más importantes ideólogos de la IA. «Los pioneros sentían intensamente que casi lo habían logrado —dice Moravec—. Creían que si cosas difíciles, como el ajedrez o los problemas de geometría, son tan fáciles de resolver para las máquinas, los temas de nivel inferior serían más fáciles todavía. Ahora, McCarthy piensa que lograr la Inteligencia Artificial es tarea para un Einstein



o un Newton. No creo que sea para tanto. Todo es cuestión de contar con ordenadores más veloces.»

Moravec cree que esos ordenadores más rápidos están en camino. «A comienzos de la década del 90 —dice— tendremos pequeños ordenadores relativamente baratos y capaces de ejecutar mil millones de operaciones por segundo. Según mis cálculos, mil millones de operaciones por segundo es el equivalente a la inteligencia de un colibrí. Una década después, tendremos ya máquinas capaces de un billón de operaciones por segundo, exactamente lo necesario para alcanzar el nivel de la inteligencia humana.»

— «¿Qué quiere decir?»

— «Que cualquier cosa que pueda llevar a cabo el ser humano, será capaz de hacer la máquina.»

— «¿Estar en el campo de béisbol, ver cómo la bola se aleja del bate, correr y cogerla?»

— «Sí, ¿por qué no?»

— «¿Quiere decir que será capaz de salir de esta habitación, bajar, llamar a un taxi, ir al aeropuerto, tomar el avión a Washington, leer algunos papeles y ponerse a escribir sobre, por ejemplo, inteligencia artificial?»

— «Sí», responde Moravec, titubeando sólo para consolar al visitante.

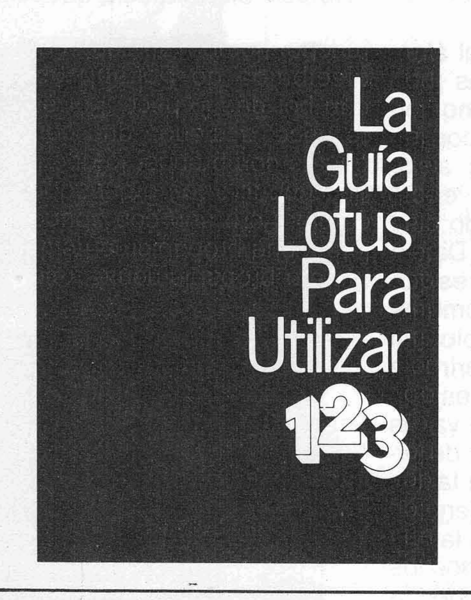
— «Tal vez —añade— algunas habilidades profesionales lleven más tiempo de reproducir.»

Actualmente, la mayoría de los investigadores en IA consideran excesivo el optimismo de Moravec. Por ejemplo, **Marvin Minsky,** del Instituto de Tecnología de Massachussetts, uno de los cuatro iniciadores de este campo de investigaciones, piensa que el ser humano podrá sentirse satisfecho si dentro de cien años llega a fabricar un robot capaz de sustituir con eficacia a un ama de casa. **John McCarthy** recuerda muy bien sus discusiones con Moravec, pero sigue pensando que una máquina genuinamente inteligente tardará mucho más en nacer que los ordenadores ultra rápidos. «Necesitaremos 1,7 Einsteins», bromea.

Queda razonablemente claro cómo resolver parte del problema de hacer un robot imaginado por Moravec, por ejemplo planear un viaje de Pittsburgh a Washington. Podría estar equipado de tablas de distancias entre determinados puntos, como ciudad a ciudad, ciudad a aeropuerto, hogar y despacho, con paradas de taxis o estaciones de ferrocarril. Y podría almacenar cantidad de conocimientos sobre los mejores me-

Lotus®

La Guía Lotus Para Utilizar 1 2 3



LA GUIA LOTUS PARA UTILIZAR 1-2-3 es un libro que le enseñará paso a paso cómo utilizar este programa.

LA GUIA LOTUS PARA UTILIZAR 1-2-3 contiene:

— Glosario detallado e índice de forma que pueda encontrar fácilmente cualquier cosa que necesite.
— Explicación de la capacidad de macros de la versión 2.
— Una biblioteca básica de macros que ofrece al nuevo usuario el descubrimiento inmediato y el uso eficiente de los macros, al mismo tiempo que aprende a programar.

CARACTERISTICAS:

- Páginas: 300
- Papel offset: 112 grs.
- Tamaño: 182 x 232 mm.
- Encuadernación: Rústica-cosido

El complemento indispensable para el manual **1-2-3**

OFERTA DE LANZAMIENTO 3.950 PTAS. (IVA INCLUIDO)

Recorte y envíe HOY MISMO este cupón a: **infodis, s.a.** c/ Bravo Murillo, 377 - 28020 MADRID

CUPON DE PEDIDO

SI. Envíenme el libro **«LA GUIA LOTUS PARA UTILIZAR 1-2-3»** al precio de **3.950** PTAS.

EL IMPORTE lo abonaré:
Con tarjeta de crédito VISA ☐ INTERBANK ☐ AMERICAN EXPRESS ☐
CONTRAREEMBOLSO ☐ ADJUNTO CHEQUE ☐
Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
Fecha de caducidad ______ Firma, ______
NOMBRE ______
DIRECCION ______
CIUDAD ______ C.P. ______
PROVINCIA ______ TELEFONO ______

TAMBIEN LO PUEDE ADQUIRIR EN SU LIBRERIA HABITUAL



dios de trasladarse dentro de distancias preestablecidas: que los aviones son preferibles para distancias superiores a, digamos, 300 kilómetros, los trenes (suponiendo que seguirán existiendo en ese entonces) para distancias entre 100 y 300 kilómetros, los automóviles para recorrer entre 8 y 100 kms., y los taxis para distancias no superiores a 8 kms. Una vez enriquecido con muchos conocimientos de este tipo, un ordenador podría establecer el plan adecuado para las necesidades de la gente promedio.

Pero, para llegar a hacer físicamente el viaje por sí mismo, necesitaría habilidades sensoriales y motrices y, sobre todo, comprender y manejarse con las inevitables circunstancias inesperadas que le acechan en el mundo real.

Finalmente, para llevar a cabo el resto de la misión, sacar consecuencias y escribir sobre sus investigaciones, el robot también tendría que tener, por lo menos, algún conocimiento sobre el millón de posibles obviedades no tan obvias para él. Algo, aunque quizás poco, del conocimiento que los hombres han venido acumulando sobre la indivisible e ininterrumpida elaboración de casi todo.

Como toda programación de ordenadores digitales, todo el edificio de la IA está construido sobre la base de varios principios de procesamiento simbólico. El primer ordenador digital, o «calculador automático», empleaba la lógica matemática para manipular símbolos numéricos y resolver ecuaciones. Pero a sus brillantes inventores no se les escapó que esas máquinas, con la misma facilidad, podían manipular símbolos no numéricos —letras, palabras, posiciones de ajedrez, mapas, diagramas— y, en general, cualquier tipo de elemento posible de ser precodificado como secuencias de números binarios (1 y 10).

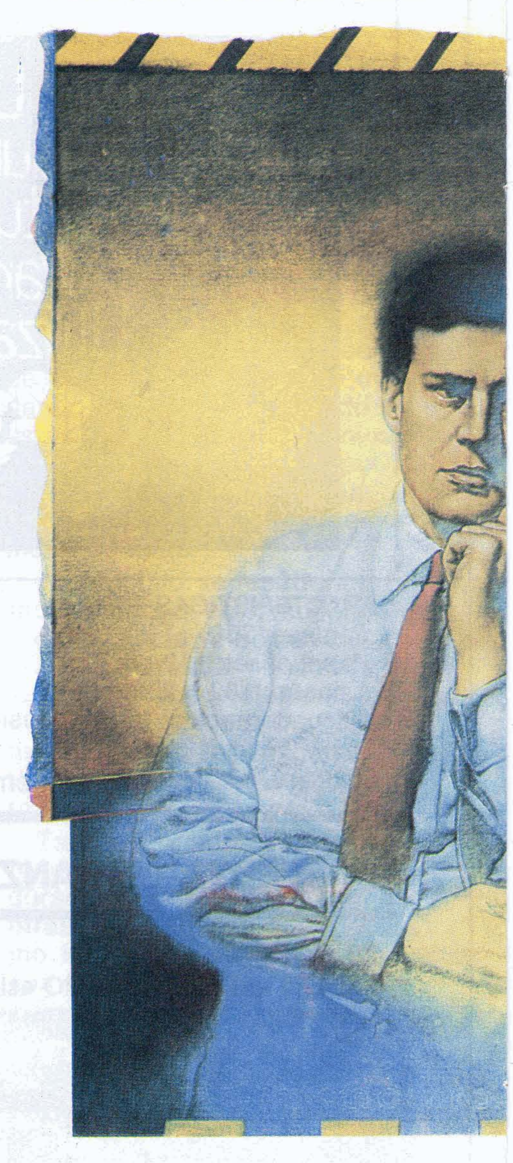
Esto abrió la perspectiva para la mecanización de la lógica no numérica o deductiva, tan reverenciada por los antiguos como sumo exponente del pensamiento humano.

Exactamente igual que el álgebra se limita a unas reglas para sustituir válidamente términos en las ecuaciones, la lógica consiste en reglas —llamadas axiomas— que sustituyen un enunciado por otro, permitiendo con ello elaborar un teorema. Dadas las premisas «Sócrates es un hombre» y «todos los hombres son mortales», por ejemplo, un simple axioma permite inferir que Sócrates es mortal. Leyes así pueden ser expresadas de varias maneras. En un programa de ordenador suelen equivaler a la forma SI-ENTONCES. Por ejemplo, «si el objeto A pertenece a la clase de los objetos B y si todos los miembros de la clase B tienen la propiedad C, entonces A tiene la propiedad C».

Esas operaciones lógicas están basadas en la habilidad del ordenador para buscar y acoplar símbolos. Si la máquina puede encontrar algún tipo de relación entre la pauta de símbolos del lado SI de la ley, puede provocar su reemplazo por los del lado ENTONCES. La lógica también depende de la habilidad de la máquina para establecer conexiones dentro de su memoria. Una importante virtud del lenguaje de programa LISP, el favorito de la mayoría de los programadores de IA, es su facilidad para crear ese tipo de estructuras de memoria. Otro lenguaje utilizado por los investigadores es el Prolog, especialmente bueno para facilitar inferencias lógicas.

En el comienzo de la IA, a finales de los años 50, los investigadores estaban tan subyugados por el poder de las máquinas que llegaron a imaginar que la aparente complejidad de la conducta inteligente podría ser reducida a alguna lógica general y espartana. Esa idea les alentó a creer que la inteligencia de las máquinas estaba al alcance de la mano.

Hace ahora 22 años, en 1965, **Herbert Simon** decía que «dentro



de 30 años, las máquinas serán capaces de realizar cualquier tarea que el hombre sea capaz de hacer». Junto con sus colegas **John Shaw** y Allen Newel, Simon escribió un programa, al que llamó Logic Theorist, para poner a prueba muchos de los teoremas de la lógica formal. Hoy se le reconoce como el primer programa de la tradición IA.

Logic Theorist y muchos de sus sucesores se basaban en leyes lógicas simples y generales, ensayando muchas combinaciones posibles de premisas simbólicas



para llegar a la demostración deseada. Pero, a medida que los investigadores van dejando atrás el artificial y austero mundo de la lógica formal por otro más realista, aquellas aproximaciones pioneras resultan hoy muy poco prácticas.

Un medio menos artificial, por ejemplo, es el juego de ajedrez. Una manera algo ingenua de enfocar el ajedrez computerizado sería alimentar al ordenador con las reglas del juego —comparables a las de la lógica— y darle instrucciones para que investigue cada posible movimiento y cada posible respuesta hasta que simplemente descubra las secuencias requeridas para llegar al juego perfecto. El problema con este método de fuerza bruta es que cada movida abre una ramificación de posibilidades para tantas respuestas y contrarréplicas que hasta al más veloz de los ordenadores le llevaría hasta el fin de los tiempos evaluar todas las posibilidades.

Por supuesto, los seres humanos, cuando juegan, solucionan este problema suplementario, las reglas formales del ajedrez con reglas informales prácticas (llamadas heurísticas, o arte de inventar) o conocimientos adquiridos por la experiencia, que sirven para limitar y orientar la búsqueda del movimiento adecuado.

Con el tiempo, la IA ha pasado a depender cada vez menos del razonamiento lógico simple y general, partiendo de los principios que los pioneros creyeron que serían suficientes. La heurística de las reglas SI-ENTONCES ha evolucionado y perfeccionado hasta llegar a un punto, el actual, en el que el virtual sinónimo de Inteligencia Artificial aplicada es la llamada «programación basada en el conocimiento». Consiste, primariamente, en sistemas que incorporan una gran cantidad de conocimientos en forma de reglas, junto con técnicas especiales para aplicar las reglas adecuadas al tiempo adecuado. Esta aproximación está compendiada por los sistemas expertos (ver Ordenador Popular de junio de este año), que son usados tanto por la industria como en el ámbito académico. Aun así, muchos investigadores universitarios, como Moravec, Minsky y McCarthy, niegan que los sistemas expertos puedan ser calificados de inteligentes en ninguno de los sentidos que se le dan a este vocablo.

Lo paradójico es que, a medida que los sistemas expertos van teniendo más éxito, tienden a ser menos inteligentes. Es decir, menos amplios, menos confiables y menos capaces de responder a situaciones imprevistas. A modo de ilustración, podemos decir que sería hazaña muy trivial redactar un programa para que un robot resuelva determinado laberinto especificándole dónde volverse a cada encrucijada. A nadie se le ocurriría calificar de inteligente a este robot, dado que una vulgar rata puede resolver el mismo laberinto basándose en sus propios errores. A diferencia de lo que ocurre con un programa, si el laberinto fuera modificado, la rata encontraría la solución. El robot, desde luego, no.

Lo que define la inteligencia es la habilidad para manejarse con los cambios y de incorporar nueva información para llegar a nuevos logros. Por esto, los teóricos de la IA, como Minsky, se desalientan al ver que muchos de los mejores colegas son atraídos por las lucrativas posibilidades de proyectos «no inteligentes» como los sistemas expertos.

No todos los investigadores de IA han sido distraídos de su trabajo, pero los que siguen adelante están muy divididos acerca de los procedimientos a seguir. Una de las divisiones fundamentales que más ha aumentado en los últimos años atañe a la lógica misma. Algunos teóricos argumentan que, al margen de que la gente actualmente piense o no lógicamente, la lógica sigue siendo la única senda posible para que la inteligencia artificial llegue a ser una disciplina susceptible de ser enseñada, programable en ordenadores digitales. Esta corriente de pensamiento tiende a estar concentrada en la Costa Oeste de Estados Unidos, bajo el liderazgo de **John McCarthy** y de **Nils J. Nilsson**, que dirige un equipo de



la firma SRI International.

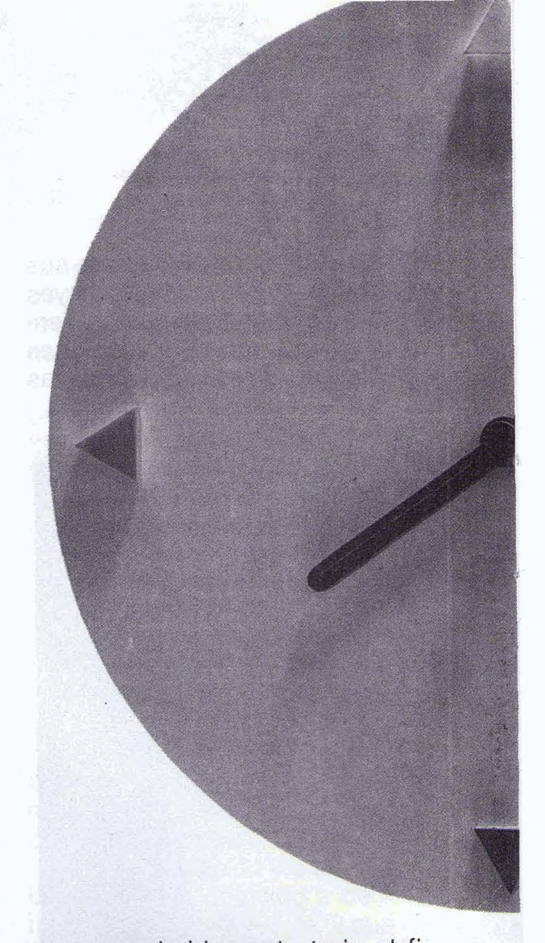
Por el contrario, en la Costa Este, teóricos como **Marvin Minsky,** del MIT, y **Robert Schank,** que trabaja en la Universidad de Yale, sostienen que la estrategia correcta consiste en descubrir y modelizar cómo se comporta la gente. Según este enfoque, la manera que la gente tiene de pensar tiene poco que ver con la lógica. Uno de los problemas de la lógica corriente es que es muy similar a las matemáticas: todo forma y nada de contenido. Cuando se infiere por lógica que Sócrates es mortal, como en el ejemplo citado, la máquina no tiene la menor sospecha de la riqueza de asociaciones que despiertan en la gente palabras como «Sócrates», «hombre» o «mortal». La escuela del Este sostiene que las máquinas tendrán que captar significados antes de ejercer la inteligencia.

Para reemplazar la lógica y las simples leyes del SI-ENTONCES, por ejemplo, Minsky estableció la noción de concepto, paquetes elaborados de conocimiento almacenado que, al igual que las asociaciones establecidas por la memoria, son evocados por conjuntos de pautas. Un concepto que describa una sintomatología dada puede contener «huecos» separados para cada síntoma y para las probables causas, efectos y tratamientos de esa enfermedad. Establecidos ciertos síntomas como palabras clave, la máquina podría hojear su colección de prototipos para encontrar la causa más plausible, las implicancias y, finalmente, el tratamiento.

La controversia sobre la lógica descuella en torno al problema de cómo lograr que los ordenadores comprendan el lenguaje «natural» humano, como por ejemplo el inglés cotidiano.

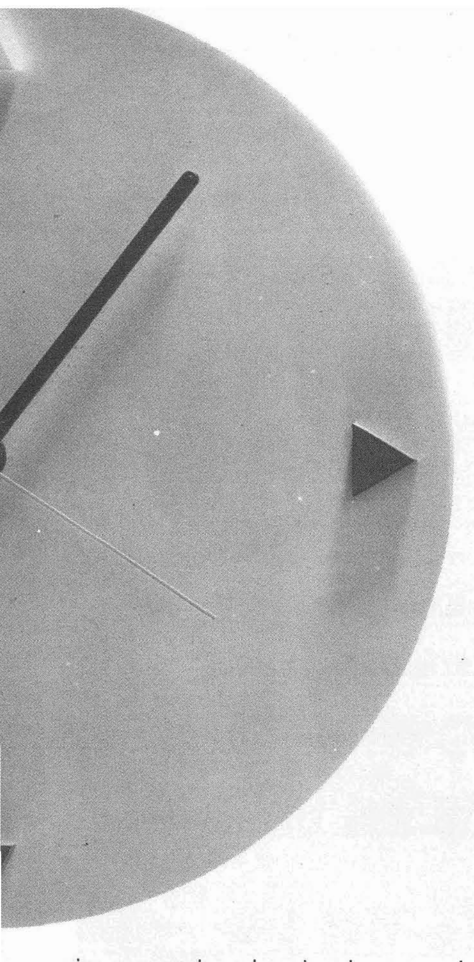
La idea imperante en la Costa Oeste norteamericana es analizar la experiencia. Este almacenamiento de conocimientos provee la información necesaria para aclarar ambigüedades y llenar los vacíos que dejan las nuevas experiencias.

Al observar que la gente pocas veces recuerda la palabra exacta para designar cosas que ha escuchado o leído, sino sólo lo esencial, Schank y sus colegas llegaron a la conclusión de que la gente traduce lo escuchado a conceptos privados, es decir que «mentaliza». Inventó, en consecuencia, una versión computada de ese proceso, al que denominó «dependencia conceptual», un lenguaje simplificado que sólo contiene once verbos en lugar de los centenares del vocabulario común del idioma inglés. Por ejemplo, su verbo sintetizado *ptrans* sirve para todas las maneras en que un objeto puede ser físicamente transportado de un lugar o dueño a otro, incluyendo movimiento, compra, vuelo y demás. La meta de la dependencia conceptual es crear pequeñas estructuras a manera de esquemas de expectativas para ayudar a la máquina a que dé un sentido a lo que lee. Encontrar un verbo tipo *ptrans* alerta a la máquina para buscar en el texto las claves de qué se mueve, cómo se mueve y cuál es su origen y destino. Una vez llenos esos huecos, el ordenador ha dado un importante paso hacia la comprensión. Schank y su equipo de Yale también han desarrollado estructuras más elaboradas de conocimientos llamadas *scripts.* Los *scripts* de **Marvin Minsky** son como miniescenarios que contienen huecos que subrayan estereotipos de experiencias. El *script* «restaurante», por ejemplo, consiste en la típica secuencia de acontecimientos con que se encuentra un típico cliente desde que entra a un restaurante hasta que sale de él. El *script* dice que Juan fue a un restaurante, pidió un filete, le pagó al camarero y salió por la puerta. El ordenador queda así capacitado para inferir que el camarero probablemente trajo el filete y que probablemente Juan lo comió. Pero aun con abundancia de *scripts,* a los ordenadores les cuesta comprender lo que la gente hace, a menos que comprendan por qué lo hace. Por eso, los discípulos de Schank pasaron mucho tiempo alimentando máquinas con otros tipos de estructuras de conocimiento, subrayando todos los objetivos humanos y las acciones típicas para satisfacer esos objetivos.



Al margen de sus excesos verbales, Schank suscita críticas entre quienes trabajan en el dominio de la IA. Muchos «pulcros» de la Costa Oeste son ingenieros dedicados *full-time* a diseñar sistemas que funcionan o a enseñar a los demás cómo hacerlo. Les sentaría muy mal que las teorías de su colega pudieran resultar acertadas.

Los sistemas semánticos necesitan enormes conjuntos de conocimientos aleatorios e hipótesis sobre cómo puede funcionar esa entidad tan compleja que es la mente humana. Los programas son muy amplios, muy *ad hoc.* Todo ese despliegue, esa necesa-



ria superabundancia de conocimientos, misceláneo que rara vez se explota, desafía a los ingenieros como una violación a sus instintos de eficacia y orden.

El portavoz del punto de vista de los ingenieros es **Nils Nilsson.** Durante años, **SRI** desarrolló más aplicaciones orientadas a la IA que cualquier otro grupo o institución. Autor de un par de libros sobre IA, Nilsson dirigió un muy publicitado proyecto cuyo fruto fue el primer robot móvil equipado con visión, Shakey, en la década de los 60. Muy posiblemente sea Nilsson y no McCarthy quien está en la mente de Schank cuando despectivamente describe a los «pulcros». Efectivamente, es atildado, con aspecto de buen mozo escandinavo, de mentalidad concisa y ordenada propia de un ingeniero, y es un locuaz defensor del enfoque lógico como representación del conocimiento.

«La gente como Schank —dice Nilsson— usa esquemas más o menos *ad hoc* que más o menos funcionan en situaciones específicas. El punto de vista de Minsky es que la inteligencia se basa realmente en esquemas unidos con chicle y alambre y que, en cierta manera, funcionan. Bueno, puede que tenga razón. Pero en ese caso no creo que sea un punto de vista demasiado inteligente. No me parece que lleguemos jamás a estar capacitados. Toda la historia de la viabilidad de ese tipo de robot se basó en tratar de simplificar principios. Creo que al construir máquinas inteligentes deberíamos, por lo menos, tener un espíritu igualmente capaz. Necesitamos encontrar principios fundamentales para ayudar a simplificar y llegar a construir ese tipo de cosas.»

Lo que, siempre según Nilsson, lleva a lo siguiente:

«¿Estaremos capacitados para emplear determinada rama de las matemáticas para comprender el mundo del sentido común que las máquinas tienen que conocer para llegar a actuar inteligentemente? Las ramas de las matemáticas más relevantes para comprender el mundo son las basadas en la lógica formal. La tradición de tratar de explicar afirmaciones sobre la vida se remonta, en lo formal, a los filósofos: Rusell, Frege y otros hicieron uso de esas técnicas de lógica formal para expresar ciertas ideas que pertenecen al campo del sentido común con mayor claridad. De hecho, toda la historia de la filosofía de este siglo se caracteriza por la tentación de acudir a los formalismos para describir lo que se estaba argumentando. Creo que la IA es un desarrollo de esto. Por ello, el tema está en si crees en la existencia de determinadas cosas tan inefables que simplemente no pueden ser formalmente descritas.»

Sea con el enfoque «zaparrastroso» o con el «pulcro», a los programadores de todo el mundo les puede llevar varias décadas equipar, por ejemplo, el robot escritor imaginado por Movarec con el conocimiento imprescindible para hacer ese viaje a Washington en el que debiera llegar a algunas conclusiones, o para educar un robot en la balística para que sea capaz de jugar aceptablemente bien al béisbol.

«Cada fragmento de conocimiento que se introduzca en un ordenador requiere una enorme codificación de información», nos dice **Roger Schank.** «Simplemente no podemos abarcar tantas codificaciones.»

Por lo tanto, son cada vez más los teóricos de la IA, incluyendo a Minsky y Schank, que llegan a la conclusión de que a los ordenadores nunca podrá llamárseles con propiedad «inteligentes» hasta que, de hecho, aprendan a aprender. Y a medida que la IA profundiza en lo que quiere decir exactamente «aprender» se hace cada vez más difícil diferenciar entre el significado de «aprendizaje» y el de «inteligencia». Aparentemente, la gente aprende por observación, análoga y generalización, comprendiendo nuevas cosas en términos ya aprendidos antes, creando siempre teorías abstractas sobre el mundo que luego les permitirán adivinar y predecir hechos de los que no tienen conocimiento directo. Este tipo de inferencia basada en la observación se llama inducción, en contraste con la deducción debida a operaciones de lógica.

Una especialista en inferencias inductivas es **Douglas Lenat,** de la Universidad de Stanford, cuyo notable programa Eurisko va mejorando poco a poco su conocimiento, comprensión y rendimiento mediante la experiencia.

Una de las características del programa es que, en vez de pensar meramente en problemas externos, puede pensar con respecto a su propio pensamiento, mediante procesos no distintos a la introspección y a la corriente de conciencia que ocupa casi siempre la mente humana. La gente revuelve casi incesantemente en su *stock* de conocimientos y observaciones, remodelando y redefiniéndolos en heurísticas cada



vez más generalizadas y en teorías con respecto al mundo.

Eurisko, efectivamente, carga «experimentos cogitados» empleando para ello un *stock* de heurística de sintaxis mediante reglas de SI-ENTONCES aplicadas a reducir frases a las partes que las componen (nombres, verbos, etc.). Una vez hecho esto, un ordenador es capaz de interpretar una orden como «¿quién es nuestro agente de ventas en Oriente Medio?» buscando en un banco de datos categorías tales como «agentes de ventas» u «Oriente Medio», y ejecutar las operaciones de acople correspondientes.

Los equipos del Este, en cambio, creen que la sintaxis es prácticamente inútil para todo menos para las comunicaciones y aplicaciones muy simples. Su esperanza se basa, en cambio, en la semántica o el significado. A su entender, los ordenadores necesitan algo mucho más parecido a las asociaciones y las imágenes mentales para resolver la ambigüedad que está casi siempre presente en el lenguaje de la gente. Por ejemplo, en una frase como «Juan llevó a Roberto al hospital después que éste se rompió la clavícula y, puesto que llovía, decidió esperar allí hasta que el médico acabara de arreglársela», el ordenador quedaría desorientado, perdido entre los pronombres y las ambigüedades semánticas, si carece de algún asidero del significado de las palabras.

Quienes proponen la semántica saben que se trata de un auténtico desafío. **Roger Schank,** que se las arregló para disgustarse con casi todos sus colegas debido a estas arrogantes declaraciones, se refiere a los grupos de las Costas Oeste y Este, respectivamente, como «los pulcros» y «los zaparrastrosos».

«La gente pulcra —dice Schank— gusta de lo formal. Usan trajes bien planchaditos y trabajan con fenómenos superficiales como la lógica y la sintaxis porque pueden verlos y comprenderlos. Los zaparrastrosos, en cambio, cuidan poco su aseo y se sienten muy felices trabajando en problemas pesados como la semántica, sólo porque es algo interesante, aunque ni remotamente ven que ello pueda aportar soluciones.»

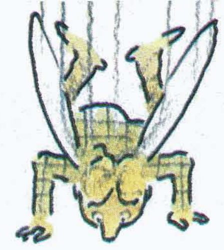


Schank, que pese a todo se identifica con este segundo campo, ve la solución del problema del lenguaje natural como equivalente a la solución de todo el problema de la IA. Antes que un ordenador pueda evidenciar una genuina inteligencia, tendrá que ser abastecido con estructuras de conocimientos elaborados y expectativas equivalentes a las que adquiere la gente como fruto de su educación e interés general para manipular «marcos» de conocimiento y «huecos». Los experimentos están guiados por reglas que deciden si un descubrimiento es «interesante» o no, ya sea que exponga regularidades inesperadas, peculiaridades o éxitos. En base a estos descubrimientos, sabe si puede añadirlos a su *stock* de heurística.

Lenat ve en los programas como Eurisko «amplificadores de inteligencia», una posible ayuda a la inteligencia humana. Son programas que pueden ayudar a la gente a pensar con respecto a dominios muy complejos aunque no extensos, sugiriendo nuevas ideas, combinándolas de diferentes maneras e identificando las más admisibles.

Durante los próximos diez o veinte años es muy posible que la IA provea varios tipos de ayuda aún más sofisticada que los limitados sistemas expertos y el programa de lenguaje natural y de visión que tanto han dado que hablar últimamente. Los próximos años veremos aparecer un aparato de lenguaje natural con programas de dar consejos, a los que la gente podrá acudir mediante discado y consultar, por ejemplo, acerca de viajes, planificación financiera, inversiones y temas médicos.

Pero hasta entonces, tendremos que pensar que el tipo de ro-



bot humanoide con el que sueña Moravec es el final de un camino que exigirá muchos años de esfuerzos y avances conceptuales, Incluso puede insumir muchos diferentes tipos de ordenadores, no iguales a los que conocemos actualmente. Ya se ha probado posible que los ordenadores llevan a cabo tareas que siguen pautas estrictas, como desarrollar sistemas expertos con estrecho margen de conocimientos y dedicados a tareas específicas. Las máquinas inteligentes de aplicación generalizada quedan, por ahora, fuera de nuestro alcance.

De lo que más obviamente carece la IA es de capacidad para entenderse fácilmente con el mundo no simbólico, sensorial, del mosquito y otras criaturas; capacidad que hasta ahora la IA ha menospreciado por considerarla inferior. Programados genéticamente o aprendidos, estos rasgos de bajo nivel son más importantes para la supervivencia biológica que otras reglas más obvias y para aprender las cuales debemos esforzarnos. Ejemplo: jugar al ajedrez.

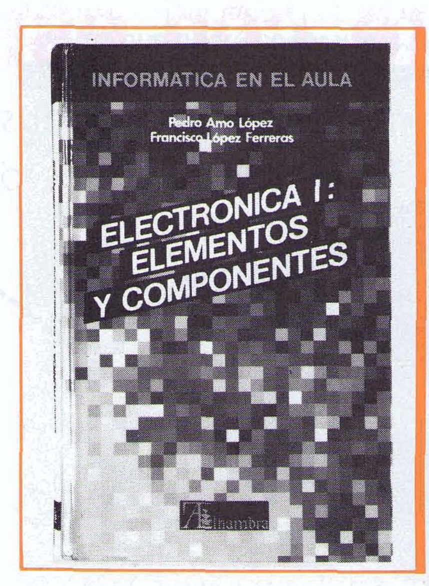
Pero las facultades sensoriales, de bajo nivel, también pueden ser componentes importantes de la inteligencia genuina. Gran parte de nuestro pensamiento, por ejemplo, adquiere la forma de imágenes mentales; es verosímil que esas imágenes deriven en determinada manera de nuestras facultades visuales tan desarrolladas que, por ahora, los ordenadores están lejos de poder emular. Además, la mayor parte de los criterios humanos parecen basarse en la intuición, en lo que se siente acertado, más que en análisis lógicos. Esas intuiciones probablemente tengan mucho más en común con el procesamiento no simbólico, paralelo, que ocurre dentro del mosquito, que con el proceso simbólico y secuencial que se lleva a cabo en el interior de los ordenadores digitales y dentro de los programas de IA tal cual los conocemos.

Así, por ahora, la inteligencia artificial sigue siendo una mera posibilidad de realización. Grandes equipos de investigación están trabajando en esa vía. Las aplicaciones prácticas derivadas de estas investigaciones pueden verse ya en la robótica, los sistemas expertos y las facilidades del lenguaje natural elemental. Pero pese a todas sus impresionantes habilidades, estas máquinas todavía no pueden pensar por sí mismas.

LA BIBLIOTECA

Electrónica:
Elementos y Componentes
Colección Informática en el Aula.
Editorial: Alhambra.
Páginas: 187 (Vol. I) y 162 (Vol. II).

Dentro de la extensa terminología propia de la informática, existen dos



palabras, muy utilizadas y que, por tanto, muchos de nuestros lectores habrán escuchado alguna vez: *Hardware* y *Software*.
En esencia, *Hardware* se refiere a los componentes físicos y tangibles del ordenador, generalmente formados por circuitos eléctricos y electrónicos (por ejemplo, un chip de memoria *ROM* es parte del *Hardware*).

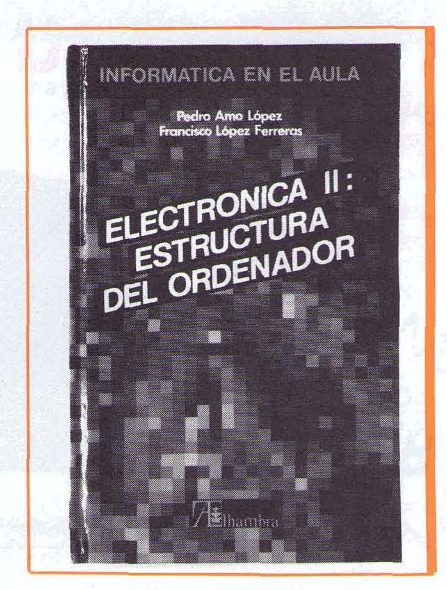
Por el contrario, *Software* se refiere a una parte del ordenador no tangible, pero igualmente importante: los programas que se ejecutan (como por ejemplo, las instrucciones contenidas en un chip de *ROM*). Algunos autores comentan que «un ordenador sin *software* es sólo un montón de chatarra».

Dentro de la colección *Informática en el Aula*, la editorial Alhambra publica esta obra en dos volúmenes dedicada a la iniciación a la electrónica digital que se utiliza en un ordenador.

Como todas las obras de esta colección, presenta un carácter marcadamente didáctico y de iniciación, aunque debido a la complejidad del tema tratado, en esta ocasión resulta interesante tener algunos conocimientos muy generales de la física para poder seguir el contenido de la obra sin grandes esfuerzos.

Los temas están agrupados en dos grandes unidades globales, correspondiendo una a cada volumen.

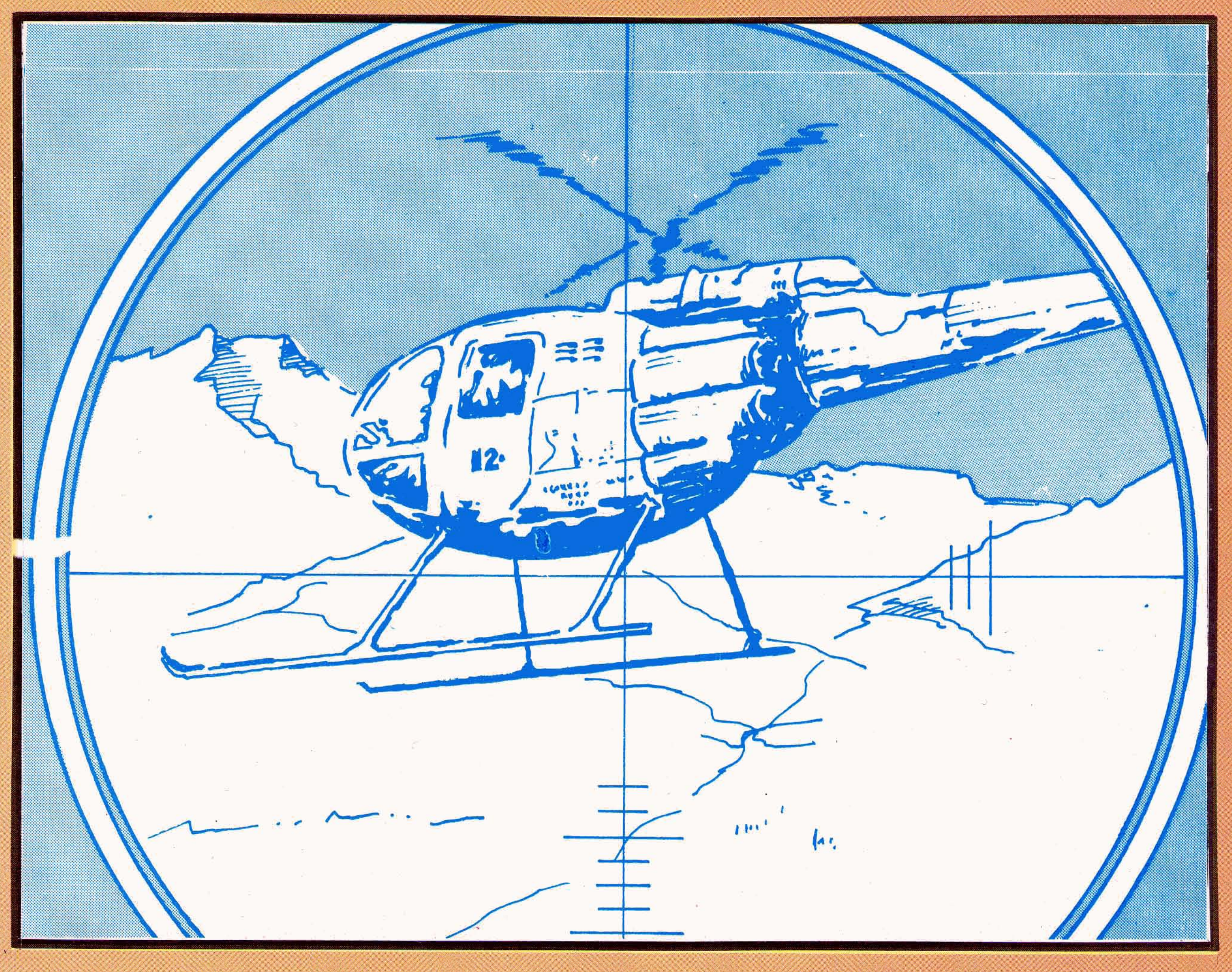
En el primero de ellos se nos introduce a la física elemental de estado sólido, necesaria para comprender someramente el funcionamiento de los materiales semiconductores y los diversos componentes electrónicos (diodos, transistores, etc.) que se fabrican a partir de ellos.
Paralelamente se nos da a conocer la lógica booleana de circuitos que utilizan los ordenadores, ya que los componentes y los circuitos electrónicos que vamos a estudiar deben ejecutar en la práctica la teoría e las citadas funciones, y por tanto debemos conocerla. También se nos enseñan los sistemas de numeración y códigos (principalmente binarios) para



que conozcamos la aritmética que utiliza el ordenador, ya que así podemos comprender los circuitos aritméticos que se estudian más adelante.
Por fin volvemos al campo de la electrónica concreta y nos introducimos en las diversas tecnologías utilizadas (bipolar, FET, JFET, MOS, CMOS, RTL,

DTL, TTL, etc.) y en los circuitos elementales (circuitos conmutadores regenerativos, biestables, monoestables, aestables, etc.) que formarán los más complejos que se estudian en el volumen segundo.
El segundo libro está dedicado a los circuitos específicos que podemos encontrar en un ordenador y la estructura general del ordenador desde una perspectiva más general.
El primer capítulo nos introduce en los diversos circuitos matemáticos y de control: semisumadores binarios, sumadores totales, sumadores en serie, multiplexores, codificadores y decodificadores, conversores de código, generadores detectores de paridad, comparadores binarios, etc.
Por el contrario, el segundo capítulo nos ayuda a conocer los circuitos secuenciales y de transmisión de datos: registros de desplazamiento, contadores asincrónicos, contadores síncronos, etc., así como los sistemas digitales y el diseño general de sistemas secuenciales.
Por último, el tercer capítulo está íntegramente dedicado a la arquitectura de ordenadores. Veamos en primer lugar un estudio funcional de la CPU, la memoria y los sistemas de Entrada/Salida, para a continuación profundizar más en cada uno de ellos por separado.
Tanto el primero como el segundo volumen se encuentran profusamente ilustrados con gráficos, tablas, esquemas, fórmulas, etc.
Además tras cada sección encontramos un resumen de lo explicado en ella y algunos ejercicios propuestos para que el lector practique, asimile y refuerce los conocimientos recién adquiridos.
De este modo, estos dos volúmenes forman una excelente obra didáctica dedicada a la iniciación del neófito en el siempre interesante mundo de la electrónica digital. Dos libros que merece la pena leer.

Invasores



Oscar Casado López nos envía desde Barcelona un juego para VIC-20 sin expansión llamado INVASORES. El juego consiste en destruir los edificios de una ciudad desierta para no chocar con ellos a medida que vas perdiendo altura. Tienes tres vidas para realizar esta azaña y poder pasar a otra nueva pantalla.

El helicóptero se mueve a una velocidad constante, siendo tu el encargado de tirar las bombas que destruirán los edificios. Esto lo consigues pulsando la barra espaciadora en el momento oportuno, única tecla que tienes que manejar para dominar este juego.

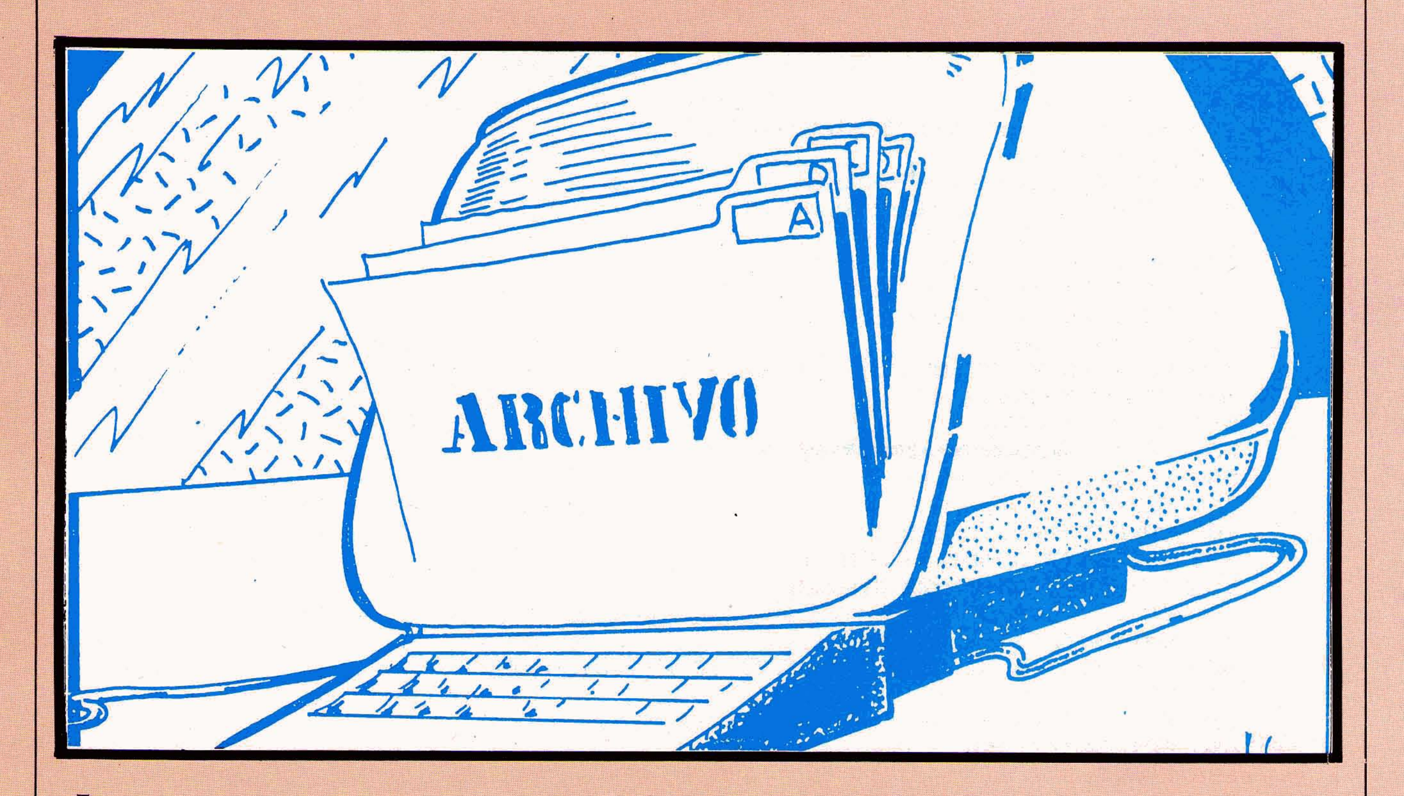
La estructura del juego es la siguiente:

1	Envía a subrutina de caracteres.
5-6	Variables.
10-380	Formación de la pantalla y de los edificios.
385-403	Movimiento del helicóptero.
405	Comprueba el choque
410	Comprueba el disparo.
420	Comprueba el paso de pantalla.
500-580	Subrutina de choque.
600-798	Subrutina de disparo.
800	Subrutina de paso de pantalla.
1000-2005	Subrutina de formación de caracteres.

```
0 REM**(C) OSCAR CASADO**
1 POKE36869,255:GOSUB1000
5 AS=0:SC=0:Q=0:II=3:S=0
6 AS=AS+1:POKE36878,15
10 A=0:B=0:PRINT"]=":POKE36879,107
20 PRINT"SQQQQQQQQQQQQQQQQQQQQR▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
30 PRINT"QPUNTOS:";SC
35 PRINT"QRECORD:";HS
36 PRINT"QP:";AS;"  V:";II;
40 B=INT(RND(1)*12)+1
50 FORI=18TOBSTEP-1
60 POKE7680+1+22*I,7
70 POKE7680+2+22*I,7
80 NEXTI
90 B=INT(RND(1)*12)+1
100 FORI=18TOBSTEP-1
110 POKE7680+4+22*I,7
120 POKE7680+5+22*I,7
130 NEXTI
140 B=INT(RND(1)*12)+1
150 FORI=18TOBSTEP-1
160 POKE7680+7+22*I,7
170 POKE7680+8+22*I,7
180 NEXTI
190 B=INT(RND(1)*12)+1
200 FORI=18TOBSTEP-1
210 POKE7680+10+22*I,7
220 POKE7680+11+22*I,7
230 NEXTI
240 B=INT(RND(1)*12)+1
250 FORI=18TOBSTEP-1
260 POKE7680+13+22*I,7
270 POKE7680+14+22*I,7
280 NEXTI
290 B=INT(RND(1)*12)+1
300 FORI=18TOBSTEP-1
310 POKE7680+16+22*I,7
320 POKE7680+17+22*I,7
330 NEXTI
340 B=INT(RND(1)*12)+1
350 FORI=18TOBSTEP-1
360 POKE7680+19+22*I,7
370 POKE7680+20+22*I,7
380 NEXTI
385 B=0
390 A=A+1:B=B+1
400 POKE7681+A,40:POKE7682+A,42:POKE7683+A,41
401 POKE36874,223
```

```
402 FORI=1TO75:NEXT:POKE7681+A,32
403 POKE36874,0
405 IFPEEK(7684+A)=7THEN500
410 IFPEEK(197)=32ORQ=1THEN600
420 IF7680+A=8094THEN800
425 GOTO390
500 II=II-1:IFII<1THENS=1
510 POKE7681+A,1:POKE7682+A,2:POKE7683+A,3
520 FORI=1TO50
530 POKE36877,255:NEXTI:POKE36877,0
535 IFS<>1THEN10
540 SC=0:A=0:B=0
550 PRINT"SQQQQQQQQQQ]]]]]]RGAME OVER"
560 FORI=1TO5000:NEXT
580 GOTO5
600 POKE7704+B,32:B=B+21
610 IFPEEK(7705+B)=7THEN670
615 IFPEEK(7705+B)=230THEN650
620 POKE7705+B,5
630 Q=1:GOTO390
650 POKE7681+A,32:POKE7704+B,230:B=A:Q=0:GOTO390
670 SC=SC+10:IFSC>HSTHENHS=SC
680 POKE7705+B,9:FORI=1TO100:NEXT:POKE7705+B,32
685 POKE36877,255
686 POKE36877,0
790 PRINT"SQQQQQQQQQQQQQQQQQQQQQRPUNTOS:";SC
795 PRINT"RRECORD:";HS
796 PRINT"RP:";AS;"  V:";II;
797 B=A:Q=0
798 GOTO390
800 Q=0:GOTO6
1000 DATA1,26,21,96,142,177,136,116,3
1001 DATA2,8,55,192,44,254,198,57,193
1002 DATA3,96,166,41,145,237,64,58,138
1003 DATA5,0,60,24,60,60,60,60,24
1004 DATA40,0,0,0,32,63,0,0,0
1005 DATA42,255,6,63,31,255,63,16,255
1006 DATA41,224,0,224,176,252,248,132,252
1007 DATA32,0,0,0,0,0,0,0,0
1008 DATA7,255,129,129,129,129,129,129,129
1009 DATA8,255,255,255,255,255,255,255,255
1010 DATA9,0,115,79,210,187,157,73,30
2000 G1=32768:G2=7168:FORI=1TO511
2001 POKEG2+I,PEEK(G1+I)
2002 NEXTI
2003 FORI=1TO11:READC:FORJ=0TO7:READK
2004 POKEG2+(C*8)+J,K:NEXTJ,I:POKE52,28:POKE56,28
2005 RETURN
```

Biblio



Juan Valera Muñoz desde Almería ha sido seleccionado por su programa BIBLIO, gestión de los libros de una biblioteca familiar para el C-64.

Según nos cuenta en su carta Juan, el programa es una escusa para proponernos algunos trucos de programación que podemos incluir en muchos de nuestros programas.

MENUS SOFISTICADOS-Las opciones se pueden elegir de dos formas:

Pulsando la tecla que corresponde a la inicial de la opción a elegir.

Desplazando la línea en vídeo inverso con las teclas de cursor y pulsando a continuación la tecla RETURN.

RUTINA DE CLASIFICACION DE CADENAS EN C.M.-Esta rutina es reubicable. Juán la ha situado al final de la zona COOO-CFFF (49152-53247) para no interferir la posible utilización de programas turbo que normalmente se colocan en esta zona. La rutina se llama con SYS 53100 y el array a clasificar se ha de llamar a$ () y a de ser unidimensional.

CLASIFICAR LOS CAMPOS DE UN REGISTRO-El método utilizado consiste en añadir en la introducción de datos el número de registros a cada campo. Completando antes con espacios en blanco la longitud máxima de cada campo. Después de la ordenación por un campo cualquiera se pasa a la variables X () el valor de los cuatro caracteres de la derecha, que es el número de orden de un registro.

A la hora de imprimirlos, sólo hay que hacer el PRINT de los caracteres de la izquierda que dimos como longitud máxima, poniendo en el índice el valor que tenga la variable X () que tiene a su vez como indice el del bucle FOR NEXT.

Este procedimiento ocupa un poco más de memoria, pero se consigue hacer la órdenación por cualquier campo.

En cuanto a la capacidad máxima de registros, se han fijado en 200, siempre queda la posibilidad de hacer diferentes ficheros agrupando los libros por temas.

```
10 REM *** BIBLIO  ***
20 REM *** J.V. SOFT 1986 ***
30 POKE53280,0:POKE53281,0:PRINT"■"
40 PRINT"■ ╭──────────────────────────────────────╮"
50 FORI=1TO22:PRINTTAB(1)"|"TAB(38)"|":NEXT
60 PRINT" ╰──────────────────────────────────────╯"
70 PRINT"■■■■■■"
80 PRINTTAB(9)"┬     ┬ ┬    ┬   ┬ ╭──╮"
90 PRINTTAB(9)"|     | |    |   | |  |"
100 PRINTTAB(9)"|     | |    |   | |  |"
110 PRINTTAB(9)"├──╮ | ├──╮ |   | |  |"
120 PRINTTAB(9)"|  | | |  | |   | |  |"
130 PRINTTAB(9)"|  | | |  | |   | |  |"
140 PRINTTAB(9)"╰──╯ ┴ ╰──╯ ╰── ┴ ╰──╯"
150 PRINTTAB(8)"■■COPYRIGHT J.V. SOFT 1987"
160 IFPEEK(53100)=32THEN180
170 GOSUB4010
180 PRINTTAB(6)"■■■■■PULSA UNA TECLA PARA EMPEZAR"
190 GETA$:IFA$=""THEN190
200 POKE53280,8:POKE53281,8:PRINT"■"
210 DIM A$(200),T$(200),AU$(200),NO$(200),X(200),F$(50)
220 REM MENU PRNCIPAL
230 C$="BIBLIO":GOSUB2820
240 P$="ELIGE UNA OPCION":GOSUB2890
250 NU=7:M$="MENU PRINCIPAL"
260 L$(1)="CREAR FICHERO":L$(2)="INTRODUCIR FICHAS":L$(3)="MODIFICAR FICHAS"
270 L$(4)="BUSCAR FICHAS":L$(5)="GRABAR FICHERO":L$(6)="LEER FICHERO"
280 L$(7)="SALIDA POR IMPRESORA"
290 GOSUB3200
300 ONOGOTO310,370,670,1130,1630,1980,2330
310 REM CREAR FICHERO
320 C$="CREAR FICHERO":GOSUB2820
330 IFFC=1THENGOSUB3570:FC=0:GOTO310
340 INPUT"■■NOMBRE DEL FICHERO:";NF$
350 FC=1
360 GOTO3530
370 REM ABRIR FICHAS
380 C$="INTRODUCIR FICHAS":GOSUB2820
390 P$="ESCRIBE FIN PARA TERMINAR":GOSUB2890
400 POKE214,4:SYS58640
410 IFFC=0THEN PRINT"ANTES DEBES CREAR EL FICHERO.":GOTO3530
420 IFCC=200THENPRINT"YA TIENES 200 FICHAS":GOTO3530
430 F=CC+1:F$=MID$(STR$(F),2)
440 FORI=LEN(F$)TO3:F$=" "+F$:NEXT
450 PRINT"HAY"CC"FICHAS"
460 PRINT"PUEDES INTRODUCIR HASTA 200 FICHAS■■"
470 PRINT"■TITULO (MAX.35 CARACTERES)"
480 INPUTT$(F)
490 IFT$(F)="FIN"THEN220
500 IFLEN(T$(F))>35THENPRINT"■■":GOTO470
510 IFLEN(T$(F))=35THEN530
520 FORI=LEN(T$(F))TO34:T$(F)=T$(F)+" ":NEXT
530 T$(F)=T$(F)+F$
540 PRINT"■AUTOR (MAX.35 CARACTERES)"
550 INPUTAU$(F)
560 IFLEN(AU$(F))>35THENPRINT"■■":GOTO540
570 IFLEN(AU$(F))=35THEN590
```

PROGRAMAS

```
580 FORI=LEN(AU$(F))TO34:AU$(F)=AU$(F)+" ":NEXT
590 AU$(F)=AU$(F)+F$
600 PRINT"■TOMO N. (MAX. 4 CARACTERES)
610 INPUTNO$(F)
620 IFLEN(NO$(F))>4THENPRINT"⊤⊤":GOTO600
630 IFLEN(NO$(F))=4THEN650
640 FORI=LEN(NO$(F))TO3:NO$(F)=" "+NO$(F):NEXT
650 NO$(F)=NO$(F)+F$
660 CC=CC+1:GOTO370
670 REM MODIFICAR FICHAS
680 C$="MODIFICAR FICHAS":GOSUB2820
690 IFCC=0THENPRINT"■■FICHERO VACIO":GOTO3530
700 Y=0
710 PRINT"■■ESCRIBE EL TITULO DE LA FICHA"
720 INPUTTT$
730 FORI=LEN(TT$)TO34:TT$=TT$+" ":NEXT
740 FORI=1TOCC
750 IFTT$=LEFT$(T$(I),35)THENY=Y+1:F$(Y)=RIGHT$(T$(I),4):X(Y)=VAL(F$(Y))
760 NEXT
770 P$(1)="MODIFICAR OTRA FICHA"
780 P$(2)="VOLVER AL MENU PRINCIPAL"
790 P$(3)=""
800 IFY<>0THEN840
810 PRINT"■■NO HAY NINGUNA FICHA CON ESE TITULO"
820 VI=1:NU=2:GOSUB2960
830 ONOGOTO 670,220
840 P$(3)="FICHA SIGUIENTE":VI=0:NU=3
850 P$="MODIFICA Y/O PULSA RETURN"
860 FORK=1TOY
870 IFK=YTHENP$(3)="":VI=1:NU=2
880 PRINT"⊐":GOSUB2820:GOSUB2890
890 IFK=1THENPOKE214,5:SYS58640:PRINT"HAY"Y"FICHAS CON ESE  TITULO"
900 POKE214,5:SYS58640
910 PRINT"■TITULO :":PRINT"■■"LEFT$(T$(X(K)),35)
920 PRINT"⊤⊤TITULO :":INPUTTT$
930 IFLEN(TT$)>35THENPRINT"⊤⊤":GOTO910
940 IFLEN(TT$)=35THEN960
950 FORI=LEN(TT$)TO34:TT$=TT$+" ":NEXT
960 T$(X(K))=TT$+F$(K)
970 PRINT"■AUTOR  :":PRINT"■■"LEFT$(AU$(X(K)),35)
980 PRINT"⊤⊤AUTOR  : ":INPUTTT$
990 IFLEN(TT$)>35THENPRINT"⊤⊤":GOTO970
1000 IFLEN(TT$)=35THEN1020
1010 FORI=LEN(TT$)TO34:TT$=TT$+" ":NEXT
1020 AU$(X(K))=TT$+F$(K)
1030 PRINT"■TOMO N.: ■■"LEFT$(NO$(X(K)),4)
1040 PRINT"⊤TOMO N.: ";:INPUTTT$
1050 IFLEN(TT$)>4THENPRINT"⊤⊤":GOTO1030
1060 IFLEN(TT$)=4THEN1080
1070 FORI=LEN(TT$)TO3:TT$=" "+TT$:NEXT
1080 NO$(X(K))=TT$+F$(K)
1090 PRINT"■■QUEDAN"Y-K"FICHAS CON ESE TITULO"
1100 GOSUB2960
1110 ONOGOTO670,220,1120
1120 NEXT
1130 REM BUSCAR FICHAS
1140 C$="BUSCAR FICHAS":GOSUB2820
```

```
1150 IFCC=0THENPRINT"FICHERO VACIO":GOTO3530
1160 P$="ELIGE UNA OPCION":GOSUB2890
1170 NU=3:M$="DEL MISMO: ":L$(1)="TITULO":L$(2)="AUTOR":L$(3)="VOLVER AL MENU"
1180 GOSUB3200
1190 ONOGOTO1200,1370,220
1200 FT=1:C$="BUSCAR FICHAS DEL MISMO TITULO":GOSUB2820
1210 Y=0
1220 PRINT"ESCRIBE EL TITULO"
1230 INPUTTT$
1240 FORI=LEN(TT$)TO34:TT$=TT$+" ":NEXT
1250 FORI=1TOCC
1260 IFTT$=LEFT$(T$(I),35)THENY=Y+1:F$(Y)=RIGHT$(T$(I),4):X(Y)=VAL(F$(Y))
1270 NEXT
1280 G$="AUTOR ":G1$="TITULO:"
1290 P$(1)="BUSCAR OTRA FICHA"
1300 P$(2)="VOLVER AL MENU PRINCIPAL"
1310 P$(3)=""
1320 IFY<>0THEN1360
1330 PRINT"NO HAY NINGUNA FICHA CON ESE TITULO"
1340 VI=1:NU=2:GOSUB2960
1350 ONOGOTO 1130,220
1360 P$(3)="FICHA SIGUIENTE":VI=0:NU=3:GOTO1540
1370 FT=0:C$="BUSCAR FICHAS DEL MISMO AUTOR":GOSUB2820
1380 Y=0
1390 PRINT"ESCRIBE EL AUTOR"
1400 INPUTTT$
1410 FORI=LEN(TT$)TO34:TT$=TT$+" ":NEXT
1420 FORI=1TOCC
1430 IFTT$=LEFT$(AU$(I),35)THENY=Y+1:F$(Y)=RIGHT$(AU$(I),4):X(Y)=VAL(F$(Y))
1440 NEXT
1450 G$="TITULO":G1$="AUTOR:"
1460 P$(1)="BUSCAR OTRA FICHA"
1470 P$(2)="VOLVER AL MENU PRINCIPAL"
1480 P$(3)=""
1490 IFY<>0THEN1530
1500 PRINT"NO HAY NINGUNA FICHA DE ESE AUTOR"
1510 REM VI=1:NU=2:GOSUB2960
1520 ONOGOTO 1130,220
1530 P$(3)="FICHA SIGUIENTE":VI=0:NU=3:GOTO1540
1540 CL=1:GOSUB3700
1550 FORK=1TOY
1560 IFCL=10THENP$="PULSA UNA TECLA PARA CONTINUAR":CL=1:GOSUB3660:GOSUB3700
1570 IFFT=1THENPRINTLEFT$(AU$(X(K)),35);LEFT$(NO$(X(K)),4)
1580 IFFT=0THENPRINTLEFT$(T$(X(K)),35);LEFT$(NO$(X(K)),4)
1590 CL=CL+1
1600 NEXT
1610 P$(3)="":VI=1:NU=2:GOSUB2960
1620 ONOGOTO1130,220
1630 REM GRABAR FICHERO
1640 C$="GRABAR FICHERO":GOSUB2820
1650 IFCC=0THENPRINT"FICHERO VACIO":GOTO3530
1660 P$="ELIGE UNA OPCION":GOSUB2890
1670 NU=3:M$="GRABAR EN:":L$(1)="CINTA":L$(2)="DISCO":L$(3)="VOLVER AL MENU"
1680 GOSUB3200
1690 ONOGOTO1700,1840,220
1700 REM GRABAR EN CINTA
1710 PRINT"":GOSUB2820:PRINT""
```

PROGRAMAS

```
1720 OPEN1,1,1,NF$
1730 PRINT#1,FC
1740 PRINT#1,NF$
1750 PRINT#1,CC
1760 FORI=1TOCC
1770 PRINT#1,T$(I)
1780 PRINT#1,AU$(I)
1790 PRINT#1,NO$(I)
1800 NEXT
1810 CLOSE1
1820 PRINT"QQFICHERO "NF$" GRABADO":GOTO3530
1830 REM GRABAR EN DISCO
1840 PRINT"J":GOSUB2820:PRINT"QQQQ"
1850 OPEN15,8,15
1860 OPEN5,8,5,"@0:"NF$",S,W"
1870 PRINT#5,FC
1880 PRINT#5,NF$
1890 PRINT#5,CC
1900 FORI=1TOCC
1910 PRINT#5,T$(I)
1920 PRINT#5,AU$(I)
1930 PRINT#5,NO$(I)
1940 NEXT
1950 CLOSE15:CLOSE5
1960 PRINT"QQFICHERO "NF$" GRABADO":GOTO3530
1970 GOSUB4010
1980 REM LEER FICHERO
1990 C$="LEER FICHERO":GOSUB2820
2000 IFCC<>0THENGOSUB3570:GOTO3570
2010 P$="ELIGE UNA OPCION":GOSUB2890
2020 NU=3:M$="LEER DE:":L$(1)="CINTA":L$(2)="DISCO":L$(3)="VOLVER  AL MENU"
2030 GOSUB3200:PRINT:PRINT
2040 ONOGOTO2050,2200,220
2050 REM LEER DE CINTA
2060 PRINT"J":GOSUB2820:PRINT"QQQQ"
2070 OPEN1,1,0,NF$
2080 INPUT#1,FC
2090 INPUT#1,NF$
2100 INPUT#1,CC
2110 FORI=1TOCC
2120 INPUT#1,T$(I)
2130 INPUT#1,AU$(I)
2140 INPUT#1,NO$(I)
2150 NEXT
2160 CLOSE1
2170 GOSUB3950:FC=1
2180 PRINT"QQFICHERO "NF$" LEIDO":GOTO3530
2190 REM LEER DE DISCO
2200 PRINT"J":GOSUB2820:PRINT"QQQQ"
2210 OPEN15,8,15
2220 OPEN5,8,5,"0:"NF$",S,R"
2230 INPUT#5,FC
2240 INPUT#5,NF$
2250 INPUT#5,CC
2260 FORI=1TOCC
2270 INPUT#5,T$(I)
2280 INPUT#5,AU$(I)
```

```
2290 INPUT#5,NO$(I)
2300 CLOSE15:CLOSE5
2310 GOSUB3950:FC=1
2320 PRINT"■■FICHERO "NF$" GRABADO":GOTO3530
2330 REM SALIDA POR IMPRESORA
2340 OPEN4,4
2350 C$="SALIDA POR IMPRESORA":GOSUB2820
2360 IFCC=0THENPRINT"■■FICHERO VACIO.":GOTO3530
2370 P$="ELIGE UN CAMPO PARA ORDENAR":GOSUB2890
2380 NU=4:L$(1)="TITULO":L$(2)="AUTOR":L$(3)="N. DEL TOMO"
2390 L$(4)="VOLVER AL MENU"
2400 M$="MENU DE CAMPOS":GOSUB3200
2410 P$="IMPRIMIENDO":GOSUB2890
2420 CP=1
2430 ONOGOTO2450,2570,2690,2440
2440 CLOSE4:GOTO220
2450 FORI=1TOCC:A$(I)=T$(I):NEXT
2460 SYS53100,A$(1)
2470 FORI=1TOCC:X(I)=VAL(RIGHT$(A$(I),4)):NEXT
2480 GOSUB3780
2490 CL=12
2500 FORK=1TOCC
2510 IFCL=62THENPRINT#4,CHR$(12):GOSUB3880
2520 PRINT#4,LEFT$(T$(X(K)),35);" ";
2530 PRINT#4,LEFT$(AU$(X(K)),35);" ";
2540 PRINT#4,LEFT$(NO$(X(K)),4)
2550 CL=CL+1:NEXT
2560 PRINT#4,CHR$(12):CLOSE4:GOTO2330
2570 FORI=1TOCC:A$(I)=AU$(I):NEXT
2580 SYS53100,A$(1)
2590 FORI=1TOCC:X(I)=VAL(RIGHT$(A$(I),4)):NEXT
2600 GOSUB3780
2610 CL=12
2620 FORK=1TOCC
2630 IFCL=62THENPRINT#4,CHR$(12):GOSUB3880
2640 PRINT#4,LEFT$(T$(X(K)),35);" ";
2650 PRINT#4,LEFT$(AU$(X(K)),35);" ";
2660 PRINT#4,LEFT$(NO$(X(K)),4)
2670 CL=CL+1:NEXT
2680 PRINT#4,CHR$(12):CLOSE4:GOTO2330
2690 FORI=1TOCC:A$(I)=NO$(I):NEXT
2700 SYS53100,A$(1)
2710 FORI=1TOCC:X(I)=VAL(RIGHT$(A$(I),4)):NEXT
2720 GOSUB3780
2730 CL=12
2740 FORK=1TOCC
2750 IFCL=62THENPRINT#4,CHR$(12):GOSUB3880
2760 PRINT#4,LEFT$(T$(X(K)),35);" ";
2770 PRINT#4,LEFT$(AU$(X(K)),35);" ";
2780 PRINT#4,LEFT$(NO$(X(K)),4)
2790 CL=CL+1:NEXT
2800 PRINT#4,CHR$(12):CLOSE4:GOTO2330
2810 GOTO2810
2820 REM RUTINA DE PRSENTACION DE CABECERAS
2830 PRINT"■ ■╭──────────────────────────────────────╮"
2840 PRINT" ■│                                      │"
2850 PRINT" ■╰──────────────────────────────────────╯"
```

PROGRAMAS

```
2860 H=20-LEN(C$)/2
2870 POKE214,1:SYS58640:PRINTTAB(H)"■"C$
2880 RETURN
2890 REM RUTINA DE PRESENTACION DE PIES
2900 H=20-LEN(P$)/2
2910 POKE214,21:SYS58640:PRINT" ■╭──────────────────────────────────────╮"
2920 PRINT" ■│                                      │"
2930 PRINT" ■╰──────────────────────────────────────╯"
2940 POKE214,22:SYS58640:PRINTTAB(H)"■"P$
2950 RETURN
2960 REM RUTINA DE MENUS DE PIE Y SELECCION
2970 POKE214,19:SYS58640:PRINT" ■╭──────────────────────────────────────╮"
2980 PRINT" ■│                                      │"
2990 PRINT" ■│                                      │"
3000 PRINT" ■│                                      │"
3010 PRINT" ■╰──────────────────────────────────────╯"
3020 POKE214,20:SYS58640:PRINTTAB(2)"■"P$(1)
3030 PRINTTAB(2)"■"P$(2)
3040 PRINTTAB(2)"■"P$(3)
3050 MA$="":LM=0:FORI=1TONU:
3060 MA$=MA$+LEFT$(P$(I),1)
3070 NEXT
3080 O=1:V=21+VM:H=2
3090 POKE214,V-1:SYS58640:PRINTTAB(H)" "P$(O)
3100 GETB$:IFB$=""THEN3100
3110 IFB$="■"ANDO=1THENGOSUB3480:GOTO3100
3120 IFB$="■"THENPRINTTAB(H)"■■ "P$(O):O=O-1:PRINTTAB(H)"■■ "P$(O):GOTO3100
3130 IFB$="■"ANDO=NUTHENGOSUB3480:GOTO3100
3140 IFB$="■"THENPRINTTAB(H)"■■ "P$(O):O=O+1:PRINTTAB(H)" "P$(O):GOTO3100
3150 IFB$=CHR$(13)THENRETURN
3160 FOROO=1TONU:IFB$=MID$(MA$,OO,1)THENO=OO:RETURN
3170 NEXT
3180 GOSUB3480:GOTO3100
3190 RETURN
3200 REM RUTINA DE PRESENTACION DE MENUS Y SELECCION
3210 PU=1:MA$="":LM=0:FORI=1TONU:IFLEN(L$(I))>LMTHENLM=LEN(L$(I))
3220 MA$=MA$+LEFT$(L$(I),1)
3230 NEXT
3240 IFM$=""THEN3280
3250 IFLM<LEN(M$)THENLM=LEN(M$):GOTO3280
3260 IFLM=LEN(M$)THENLM=LEN(M$):GOTO3280
3270 FORI=1TOLM-LEN(M$):M$=M$+" ":NEXT
3280 FORI=1TONU:FORJ=1TOLM+1-LEN(L$(I)):L$(I)=L$(I)+" ":NEXT:NEXT
3290 VI=12-INT(NU/2):V=VI:H=18-INT(LM/2):VE=VI
3300 IFM$<>""THENV=VI-2:VE=V:POKE214,V:SYS58640:PRINTTAB(H)"■╭";
3310 FORI=1TOLM+2:PRINT"■─";:NEXT:PRINT"■╮"
3320 PRINTTAB(H)"■│ "M$"■ │"
3330 PRINTTAB(H)"■├";:FORI=1TOLM+2:PRINT"■─";:NEXT:PRINT"┤"
3340 FORI=1TONU:PRINTTAB(H)"■│ "L$(I)"│":NEXT
3350 PRINTTAB(H)"■╰";:FORI=1TOLM+2:PRINT"─";:NEXT:PRINT"╯"
3360 IFXX=1THENRETURN
3370 O=1:V=VI+1:H=H+1:A$=L$(1)
3380 POKE214,V:SYS58640:PRINTTAB(H)" "L$(O)
3390 GETB$:IFB$=""THEN3390
3400 IFB$="■"ANDO=1THENGOSUB3480:GOTO3390
3410 IFB$="■"THENPRINTTAB(H)"■■ "L$(O):O=O-1:PRINTTAB(H)"■■ "L$(O):GOTO3390
3420 IFB$="■"ANDO=NUTHENGOSUB3480:GOTO3390
```

```
3430 IFB$="■"THENPRINTTAB(H)"■■ "L$(O):O=O+1:PRINTTAB(H)" "L$(O):GOTO3390
3440 IFB$=CHR$(13)THENRETURN
3450 FOROO=1TONU:IFB$=MID$(MA$,OO,1)THENO=OO:RETURN
3460 NEXT
3470 GOSUB3480:GOTO3390
3480 REM ** PITIDO **
3490 VV=54272:POKE54296,15:POKE54295,0
3500 POKEVV+6,0:POKEVV+5,31:POKEVV+1,180:POKEVV+4,33
3510 FORNN=1TO100:NEXT
3520 POKEVV+4,0:RETURN
3530 REM VOLVER AL MENU PRINCIPAL
3540 P$="PULSA UNA TECLA PARA VOLVER AL MENU":GOSUB2890
3550 GETC$:IFC$=""THEN3550
3560 GOTO220
3570 REM S/R EXISTE FICHERO
3580 PRINT"■■ATENCION !!!!"
3590 PRINT"■EXISTE UN FICHERO LLAMADO: "NF$
3600 PRINT"■SI CREAS UNO NUEVO ESTE SE BORRARA."
3610 PRINT"■QUIERES SALVARLO ANTES DE CREAR OTRO?    (S/N)"
3620 GETA$:IFA$=""THEN3620
3630 IFA$<>"S"ANDA$<>"N"THENPRINT"■■■■":GOTO3610
3640 IFA$="S"THEN:GOTO1630
3650 IFA$="N"THENCC=0:RETURN
3660 REM S/R PARA MAS 8 REGISTROS EN LA BUSQUEDA
3670 P$="PULSA UNA TECLA PARA CONTINUAR":GOSUB2890
3680 GETB$:IFB$=""THEN3680
3690 RETURN
3700 REM ENCABEZ. DE BUSQUEDA
3710 PRINT"■":GOSUB2820
3720 PRINT"■■"G1$
3730 PRINTTT$
3740 PRINT"──────────────────────────────────────"
3750 PRINTTAB(10)G$SPC(13)"N.DEL TOMO"
3760 PRINT"──────────────────────────────────────"
3770 RETURN
3780 REM TITULO SALIDA IMPRESORA
3790 FI=0
3800 IFO=1THEN NN$="TITULO"
3810 IFO=2THEN NN$="AUTOR"
3820 IFO=3THEN NN$="N. DEL TOMO"
3830 PRINT#4:PRINT#4
3840 PRINT#4,CHR$(14)TAB(13)"BIBLIOTECA"
3850 PRINT#4,TAB(13)"==========";CHR$(15)
3860 PRINT#4:PRINT#4,TAB(26)"TEMA: "NF$
3870 PRINT#4:PRINT#4,"LISTA DE LIBROS ORDENADOS POR EL "NN$
3880 REM ENCABEZAMIENTO IMPRESORA
3890 IFFI=1THENPRINT#4:PRINT#4
3900 PRINT#4,TAB(69)"PAG."CP
3910 FORI=1TO76:PRINT#4,"-";:NEXT:PRINT#4
3920 PRINT#4,TAB(14)"TITULO"SPC(30)"AUTOR"SPC(13)"TOMO N."
3930 FORI=1TO76:PRINT#4,"-";:NEXT:PRINT#4
3940 FI=1:CL=6:CP=CP+1:RETURN
3950 REM S/R FORMATEO NO$ DESPUES DE LA LECTURA DEL FICHERO
3960 FORI=1TOCC
3970 IFLEN(NO$(I))=8THEN3990
3980 FORJ=LEN(NO$(I))TO7:NO$(I)=" "+NO$(I):NEXT
3990 NEXT
```

```
4000 RETURN
4010 REM S/R CLASIFICAR
4020 RESTORE
4030 I=53100
4040 READA:IFA=256THENRETURN
4050 POKEI,A:I=I+1:GOTO4040
4060 DATA32,253,174,32,139,176,133,106
4070 DATA132,107,160,0,177,106,240,122
4080 DATA133,110,200,177,106,153,110,0
4090 DATA192,2,208,246,165,106,133,108
4100 DATA165,107,133,109,24,165,108,105
4110 DATA3,133,108,144,2,230,109,160
4120 DATA0,177,108,240,71,133,252,197
4130 DATA110,144,2,165,110,133,255,200
4140 DATA177,108,153,252,0,192,2,208
4150 DATA246,160,0,177,111,209,253,240
4160 DATA4,144,209,176,9,200,196,255
4170 DATA208,241,196,110,176,198,160,0
4180 DATA177,106,170,177,108,145,106,153
4190 DATA110,0,138,145,108,200,192,3
4200 DATA208,238,160,0,177,106,133,255
4210 DATA169,0,240,168,24,165,106,105
4220 DATA3,133,106,144,2,230,107,24
4230 DATA144,128,96,256
```

CODIGOS DE CONTROL PARA EL VIC-20 Y EL C-64

Colores del VIC-20 y del 64

Cómo se ve	Cómo se teclea	Efecto conseguido
	Ctrl + 1	Negro
	Ctrl + 2	Blanco
	Ctrl + 3	Rojo
	Ctrl + 4	Cian
	Ctrl + 5	Púrpura
	Ctrl + 6	Verde
	Ctrl + 7	Azul
	Ctrl + 8	Amarillo

Colores del 64 solamente

Cómo se ve	Cómo se teclea	Efecto conseguido
	Cbm + 1	Naranja
	Cbm + 2	Marrón
	Cbm + 3	Rosa
	Cbm + 4	Gris oscuro
	Cbm + 5	Gris medio
	Cbm + 6	Verde claro
	Cbm + 7	Azul claro
	Cbm + 8	Gris claro

Códigos de cursor y control

Cómo se ve	Cómo se teclea	Efecto conseguido
	Home	Cursor a casa
	Shift + home	Limpia pantalla
	Crsr	Cursor derecha
	Shift + crsr	Cursor izquierda
	Crsr	Cursor abajo
	Shift + crsr	Cursor arriba
	Ctrl + 9	Carácter inverso
	Ctrl + 0	Carácter normal
	Del	Borrar
	Shift + del	Insertar

Teclas de función

Cómo se ve	Cómo se teclea
	F1
	F2 = Shift + F1
	F3
	F4 = Shift + F3
	F5
	F6 = Shift + F5
	F7
	F8 = Shift + F7

ESPECIAL commodore

Los mejores juegos, trucos, artículos y aplicaciones



commodore Magazine

P.V.P. 795 pts.

commodore Magazine SERVICIO



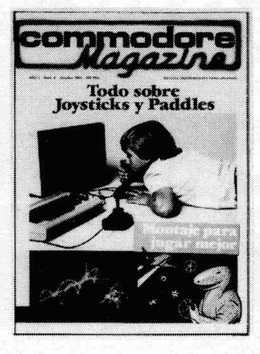
Núm. 5
Programas, juegos y concurso/Londres: Quinta feria de Commodore/Basic, versión 4.75.



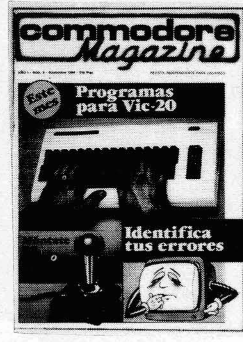
Núm. 6
El misterio del Basic/Lápices ópticos para todos/Concurso, juegos, aplicaciones.



Núm. 7
El ordenador virtuoso. MusiCalc. Programa monitor para el 64. Lápices ópticos. Ampliación de memoria para Vic-20.



Núm. 8
Joystick y Paddle para todos. Misterio del BASIC. EL LOGO. Cálculo financiero. Programas.



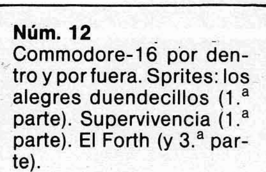
Núm. 9
Conversión de programas del Vic-20 al C-64. Móntale un paddle. Identifica tus errores. Software comentado.



Núm. 10
Koala Pad: La potencia de un paquete gráfico. Trucos. El FORTH. Software comentado. EL LOGO.



Núm. 11
Music-64. Supervivencia (1.ª parte). Cómo guarda el diskette la información. Sintetizador-64. El Forth (1.ª parte).



Núm. 12
Commodore-16 por dentro y por fuera. Sprites: los alegres duendecillos (1.ª parte). Supervivencia (1.ª parte). El Forth (y 3.ª parte).





Núm. 13
Análisis: programas de ajedrez. Los Cazafantasmas, 64. Vic en el espacio. La impresora que dibuja. Interface paralelo.



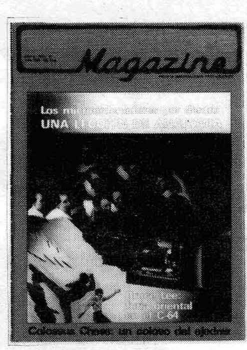
Núm. 14
Sprites; cómo entenderse con los duendes. Pilot: un lenguaje de alto nivel. Guía de Software para C-64.



Núm. 15
Síntesis de voz: su ordenador tiene la palabra. Pilot: un lenguaje de alto nivel (2.ª parte). Guía de software para C-64 (2.ª parte).



Núm. 16
Análisis de simuladores: vuela con tu C-64. Contabilidad para pequeños negocios. Cómo acelerar la ejecución de gráficos en BASIC. Submarino Commander. Pilot un lenguaje de alto nivel (3.ª parte).



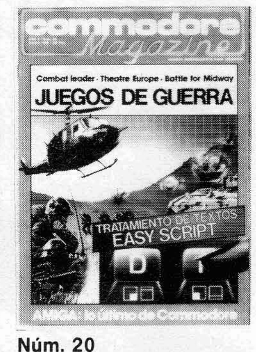
Núm. 17
Una lección de anatomía: los microordenadores por dentro. Bruce Lee: la furia oriental en el C-64. Quick Data Drive. Colossus Chess: un coloso del ajedrez.



Núm. 18
Practicalc: todo el poder de una hoja electrónica. Pascal (1.ª parte). Programas: juegos y aplicaciones veraniegos. ¡Canasta!: dos ases del baloncesto para el C-64.



Núm. 19
Análisis de cuatro impresoras. Simulación: pequeños mundos en su ordenador. Pascal (2.ª parte). Entombedy The Staff of Karnath: aventuras gráficas y mucha acción.



Núm. 20
Juegos de Guerra: Combat leader, Theatre Europe, Battle for Midway. Tratamiento de textos Easy Script. Amiga: lo último de Commodore. Libros, juegos y aplicaciones.



Núm. 21
Video-Digitizer: visión artificial para Commodore. Seikosha GP 700 VC: una impresora a todo color y con capacidades gráficas. Sprites multicolores. El nuevo C-128.

DE EJEMPLARES ATRASADOS

Complete su colección de COMMODORE MAGAZINE

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.



Núm. 22
Programas lightning: gráficos profesionales a tu alcance. Montaje: un interruptor programable para el C-64. Sprites múltiples. Cómo graba los datos el Datassette.



Núm. 25
Los ports: conexiones al exterior. El BASIC del C-128 Matemáticas por ordenador. Software educativo. Melodías musicales del Solfeo al Basic.



Núm. 27
Inteligencia Artificial: los lenguajes expertos. Sistemas de numeración: binario y hexadecimal. Comandos de disco C-128. Ficheros secuenciales. Libros, juegos y trucos.



Núm. 29
Piratas: problemas y legislación entorno a este tema. Backups: comentario del cartucho «Freeze Frame» y del Interface «Cosmos' Thoug». Introducción al PC de Commodore. Los juegos de Lucas Film.



Núm. 32
Gráficos en el Vic-20, C-64 y C-128. Hablemos de ordenadores o de cómo conseguir que no nos entienda nadie. Compatibles: fenómeno importante en el mundo de la microinformática profesional.



Núm. 23
Sinfonías en Chip: síntesis de sonido. Sprites en movimiento. Paisajes fractales en tu Commodore. Código máquina. Sight and sound: cuatro maestros de la música. Galería de Software: los mejores programas analizados en profundidad.



Núm. 26
Joysticks: ¿cómo son?, ¿cuáles son?, ¿cómo se programan?. Gráficos en código máquina. Simulador Spectrum. Proyecto Atenea.



Núm. 28
Robótica: evolución de la robótica y las posibilidades técnicas de los Robots. MECOMO: el brazo Robot. Mapa de memoria del C-128. Nuevas profesiones.



Núm. 31
La Magia del Amiga: primeras impresiones y posibilidades que este micro puede ofrecernos. Los periféricos que vienen. Mutaciones de personalidad.



Núm. 33
Reportaje fotográfico sobre la nueva imagen del C-64. Ofimática: la utilidad de los ordenadores en la empresa. Ordenadores de segunda mando. Juegos de Ingenio.



Núm. 24
El ordenador en la enseñanza. ADA para Commodore. El C-128 en fotos. Los peques y el ordenador. Seikosha Sp-1000 VC. Investigación sobre el sonido.

CORTE Y ENVIE ESTE CUPON A **COMMODORE MAGAZINE**

SERVICIO DE EJEMPLARES ATRASADOS

Bravo Murillo, 377. Tel.: 7337969 - 28020 MADRID

Ruego me envíen al precio de **300** ptas. los siguientes ejemplares atrasados de COMMODORE MAGAZINE ____________________

El importe lo abonaré

Contra reembolso ☐ Cheque adjunto ☐ Con mi tarjeta de crédito ☐ American Express

☐ Visa ☐ Interbank

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad ____________________

NOMBRE ____________________

DIRECCION ____________________

POBLACION ____________________ C.P. __________

PROVINCIA ____________________



MAPA



HABILIDAD

Starquake

Nos encontramos en el Espacio, en alguna parte de la Galaxia; un planeta desconocido está a punto de hacer que desaparezca todo el Universo, sólo se podrá evitar si el núcleo de ese pequeño planeta es reconstruido. Para evitar el desastre habrá que mandar a alguien al espacio exterior, un pequeño androide, BLOB, será el encargado de encontrar las piezas del núcleo y recomponerlo a tiempo.

Pero esto no es nada fácil, en el espacio se encontrará muchos enemigos que le robarán la energía, él podrá matarlos disparando su arma o bien, intentar esquivarlos, pero un gran laberinto de cuevas y pasadizos le esperan, en el camino encontrará objetos que podrá coger y le darán energía extra, o le recargarán sus armas, otros serán los que debe conservar para reconstruir el núcleo, y habrá uno que le servirá para acceder a los códigos secretos, para abrirle tramos prohibidos y para cambiarle unas piezas por otras que él elija.

El camino es muy largo y siempre descubrirás algo nuevo,





hay multitud de pantallas que irás recorriendo hasta conseguir todas las piezas del núcleo, para ir más deprisa podrás acoplarte un propulsor que encontrarás en varias zonas de tu recorrido, y que podrás abandonar también, cuando quieras recoger algún objeto que te interese, en esas mismas zonas.

A ver si eres capaz de ayudar a BLOB en su misión y así salvar el futuro del Universo.

Tanto los gráficos como el sonido son muy buenos, este juego es capaz de tenerte durante horas ante tu ordenador; se presenta como un reto para ti y tu curiosidad, ya que cada vez que juegues avanzarás más descubrirás, nuevos pasadizos, nuevos enemigos, pero no desesperes, te acabarás haciendo amigo de BLOB.

En general, es un juego muy original y muy bien pensando para mantener el interés por mucho tiempo del usuario.

FICHA DEL JUEGO

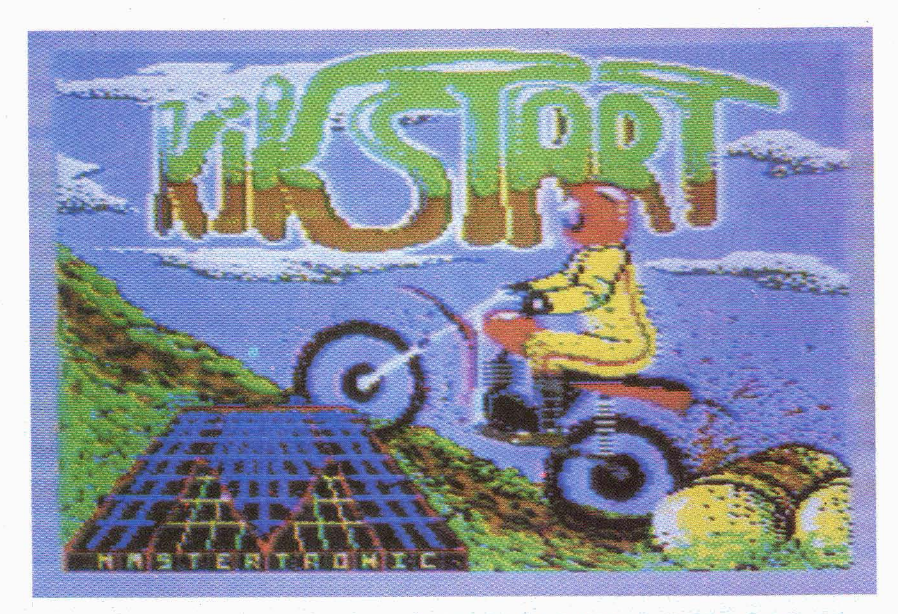
NOMBRE:	**STARQUAKE**
DE:	**MARTECH (Zafiro)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

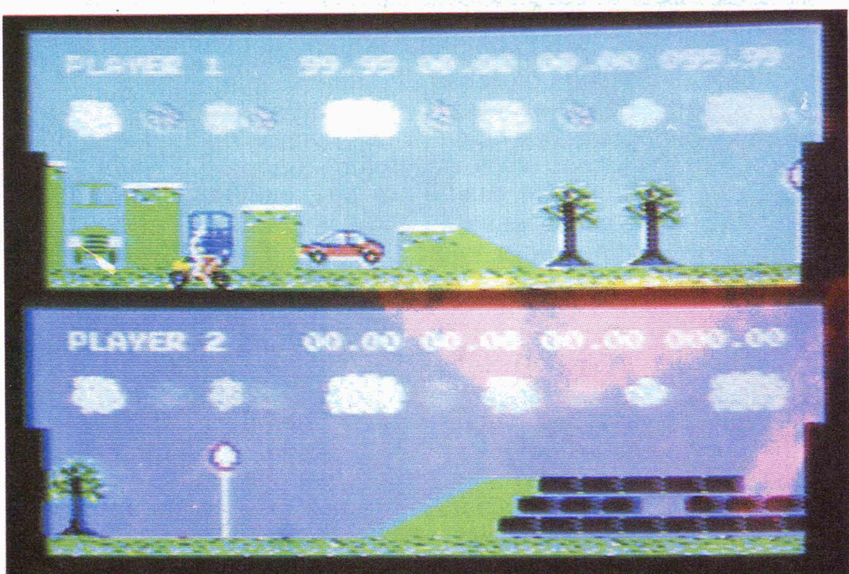
PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION									■	
GRAFICOS								■		
SONIDO								■		
GENERAL									■	





Kikstart





FICHA DEL JUEGO

NOMBRE:	**KIKSTART**
DE:	**MASTERTRONIC (Drosoft)**
ORDENADOR:	**COMMODORE-64**
CONTROL:	**JOYSTICK, TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION								■		
GRAFICOS									■	
SONIDO							■			
GENERAL								■		

Si te gustan las motos y el más difícil todavía, MASTERTRONIC te propone en un solo juego lo que buscas, a lo largo de un recorrido de ocho pistas deberás demostrar lo hábil que eres.

Durante el camino te encontrarás toda clase de obstáculos que tendrás que salvar como puedas, habrá desniveles de terreno, muros de ladrillo, vallas; tendrás que saltar sobre autobuses, filas de coches, y otros pequeños problemillas que irás descubriendo, como zonas en las que sí podrás pararte y zonas por las que deberás aumentar tu velocidad pero sin llegar a saltar, ya que eso podría ser peligroso.

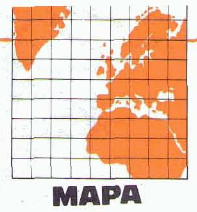
En el menú principal podrás elegir jugar con otro competidor a la vez y ver además el recorrido de ambos simultáneamente, también puedes variar el orden de los obstáculos poniéndolos en el lugar que tú elijas, puedes empezar con recorridos fáciles y complicártelos cada vez más, o si eres decidido o te crees lo suficientemente diestro con la moto, empezar con el mayor número de obstáculos posibles, todo es arriesgarse.

Es un juego entretenido, los gráficos están bien logrados, la pantalla está dividida en dos, tú jugarás en un lado y tu contrincante en el otro, así podrás ver las estrategias que él usa para salvar los obstáculos, aunque también él verá las tuyas.

Por el sonido de tu moto, sabrás si estás tomando impulso para el salto, o si estás frenando para pasar por encima de las vallas, por ejemplo.

Así que con todas estas pistas, no te queda más que insistir con las posibilidades que te ofrece este juego de convertirte en un gran motorista capaz de superar cualquier problema que se te ponga delante. Con un poco de práctica verás como consigues que tu moto escale, etc...

SOFTWARE







Explorer

Os presentamos ahora un juego, para los que tengan alma de explorador. La situación es algo peligrosa, y no se trata simplemente de hacer de *boyscout* espacial, no.

La mala fortuna, hizo que tuvieras un terrible accidente en uno de los planetas más laberínticos del Espacio Conocido. Esmeralda (que así se llama el planeta en cuestión), tiene más de cuarenta millones de posibles localizaciones, entre las cuales se encuentran las nueve piezas de tu nave, que debes recuperar.

Para moverte por Esmeralda, posees un jet portátil, equipado con un buen instrumental que será muy necesario para encontrar las piezas de tu nave.

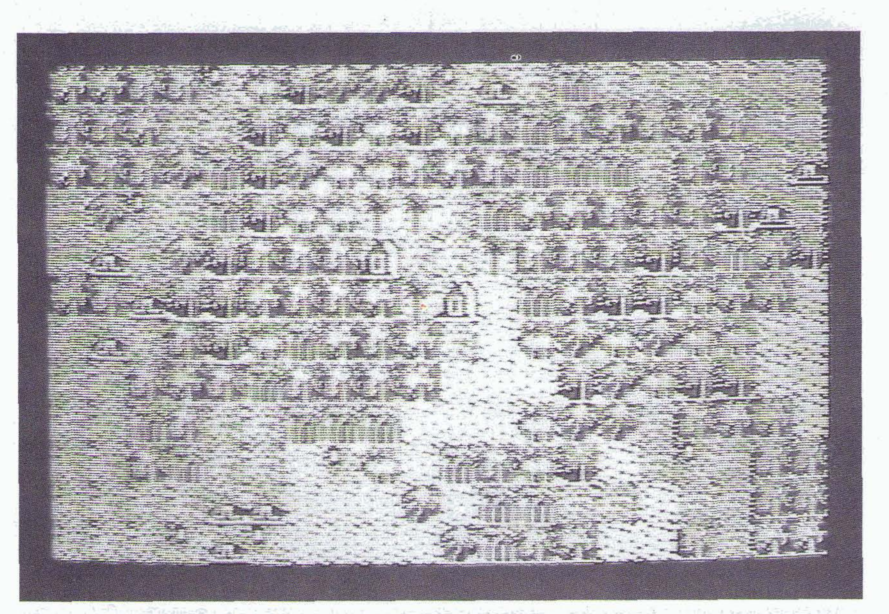
No te dejes llevar por lo atractivo del paisaje. A pesar de su aparente belleza existen en Esmeralda lugares y seres realmente peligrosos, y otros que deberás utilizar para conseguir tu objetivo. No cabe duda, si posees un buen sentido de la orientación, tu recorrido será más corto.

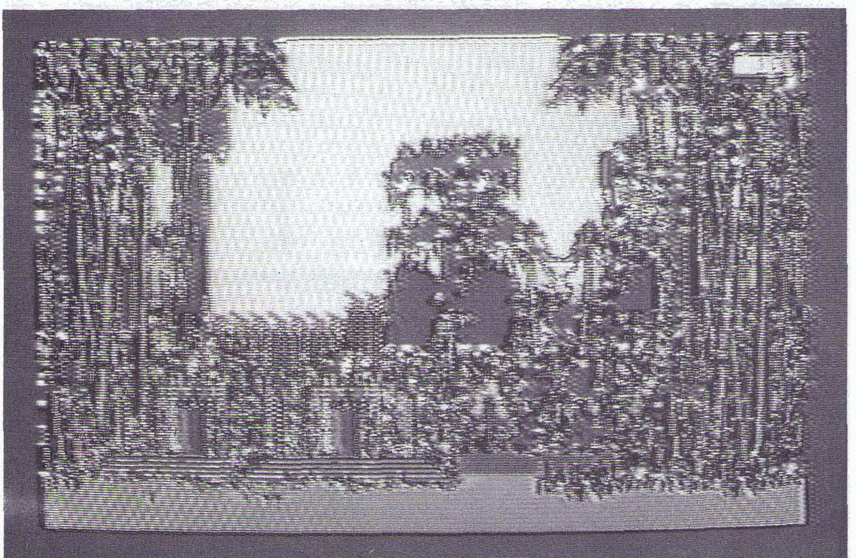
Lo cierto es que los fabricantes de tu nave, no hicieron un buen trabajo, (se estrelló en cuanto tomó contacto con la atmósfera de Esmaralda), sin embargo, el jet debe ser obra de otras personas. Te ofrece gran libertad de movimientos, y una serie de instrumentos que te serán muy útiles en tu camino por la selva.

La brújula, la pistola láser, el radiovector, el sonar de objetos... todos estos aparatos serán grandes aliados para ti en esta ocasión.

A lo largo del juego, las sorpresas y paisajes diferentes te parecerán ilimitados. Los gráficos son de una calidad tan buena, que aunque no consigas reunir las nueve piezas, te habrá merecido la pena la aventura, sólo por conocer un planeta tan singular como Esmeralda.

Debes estar muy atento. Cualquier cosa que veas, oigas o... presientas, puede ser decisiva en tu aventura. Animo, eres el único explorador, que puede lograrlo.





FICHA DEL JUEGO

NOMBRE: **EXPLORER**
DE: **ELECTRIC DREAMS (Proeinsa)**
ORDENADOR: **COMMODORE-64, 128**
CONTROL: **JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION								■		
GRAFICOS										■
SONIDO							■			
GENERAL								■		

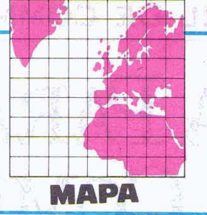
SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA.
LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A:
CONCURSO COMMODORE MAGAZINE



commodore Magazine
C/BRAVO MURILLO, 377 5.° A 28020 MADRID







Feud



Un misterioso jardín, extraños personajes... estás a punto de sumergirte en un mundo mágico y especial, en el que podrás vivir una apasionante aventura.

Hace mucho, mucho tiempo, alguien que conocía los grandes poderes de todas las magias, escribió un libro que incluía un resumen de los hechizos más poderosos.

Pues bien, nos encontramos ante una situación que tiene mucho que ver con este libro. Dos magos rivales, se dedican desde hace tiempo a lanzarse maléficos hechizos mutuamente. Uno de estos magos, eres tu, Learic, y debes vencer de una forma «mágica» al mago Leanoric, quien por su parte, no dudará en dedicarte los más terribles conjuros y maleficios que puedas imaginar.

Si hay algo realmente peligroso, es la lucha entre dos magos rivales. Llevan muchos años intentando demostrar cuál de los dos tiene más poder e ingenio al lanzar sus hechizos.

Para demostrar tu superioridad, tendrás que fijarte muy bien, en cuales son los ingredientes necesarios para poder hacer efectivo cada hechizo. el viejo libro, que antes mencionábamos, te proporcionará tan valiosa información.

En el jardín mágico, encontrarás todo lo necesario para realizar esas mezclas. Cuando creas que estás preparado para llevar a cabo el hechizo, y tengas todos los ingredientes necesarios, deberás dirigirte a tu caldero mágico y realizar la mezcla.

Cada vez que obtengas uno de los elementos necesarios para el brebaje, el libro te lo indicará dándole un color especial al nombre del ingrediente en cuestión.

Puedes pasar las páginas del libro cuando desees, y escoger con mucho cuidado el maleficio que quieres realizar.

Debes ser rápido y cauteloso... Leanoric está decidido a demostrar que él es más fuerte, y que su magia es... la más peligrosa.



FICHA DEL JUEGO

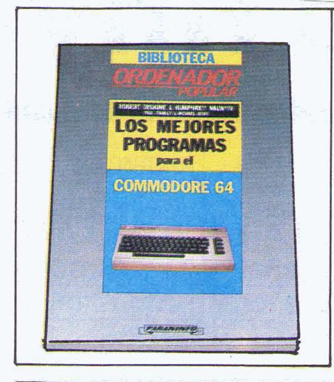
NOMBRE:	**FEUD**
DE:	**BULLDOG SOFWARE (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION							■			
GRAFICOS							■			
SONIDO						■				
GENERAL							■			

infodis, s.a. LE OFRECE LOS MEJORES LIBROS PARA SU ORDENADOR



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Descubre los misterios de la programación de una forma sencilla, con ejemplos, programas y organigramas.
(110 páginas, tamaño 13,5 x 21)



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Con utilidades, juegos explosivos y gráficos dinámicos que lleva al BASIC hasta el mejor aprovechamiento de sus posibilidades.
(200 páginas, tamaño 15,5 x 21,5).



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Un libro especialmente dedicado a los que se inician por vez primera en el mundo del Spectrum.
(100 páginas, tamaño 13,5 x 21).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Una inestimable ayuda que complementará la que proporciona el manual del ordenador.
(108 páginas tamaño 13,5 x 21,5).



P.V.P. 900 PTAS.
(IVA INCLUIDO)
Un compendio de los programas más diversos con los que podrá aprender jugando las importantnes características del BASIC.
(258 páginas, tamaño 15,5 x 21,5).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Muestra una visión más completa del correcto funcionamiento del juego de instrucciones del C-64.
(108 páginas, tamaño 13,5 x 21,5).

CUPON DE PEDIDO

enviar a:
infodis, s.a.
C/BRAVO MURILLO, 377
28020 MADRID

COPIE O RECORTE ESTE BOLETIN DE PEDIDO

DESEO RECIBIR LOS SIGUIENTES TITULOS:
15 HORAS CON EL SPECTRUM (P.V.P. 750) ☐
LOS MEJORES PROGRAMAS PARA EL ZX SPECTRUM (P.V.P. 900) ☐
LOS MEJORES PROGRAMAS PARA EL COMMODORE 64 (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL I (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL II (P.V.P. 800) ☐
(más 100 ptas. de gastos de envío).

El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐
Número de mi tarjeta: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

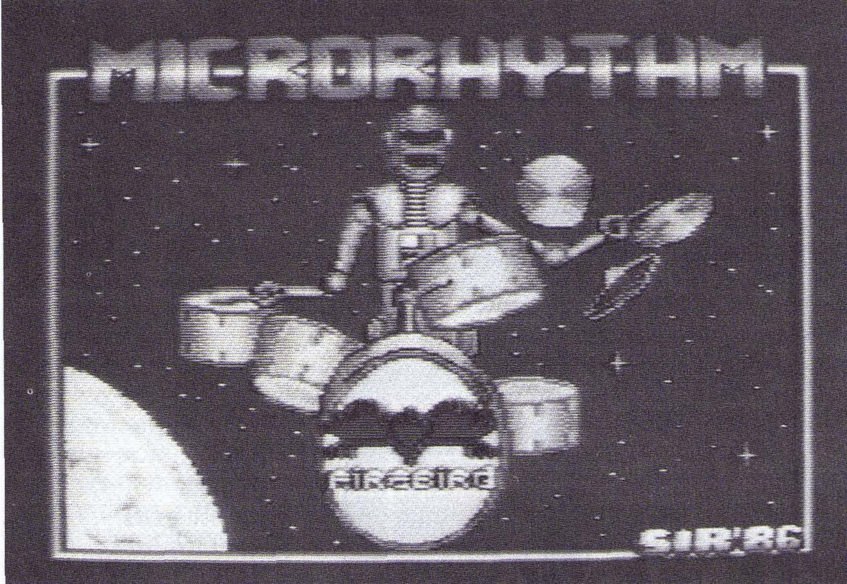
NOMBRE ____________________
CALLE ____________________
CIUDAD ____________________
PROVINCIA ____________________ C. P. __________

Micro Rhythm

FICHA DEL JUEGO

NOMBRE:	**MICRO RHYTHM**
DE:	**FIREBIRD (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION								■		
GRAFICOS						■				
SONIDO										■
GENERAL								■		



¿Por qué empeñarnos en jugar siempre a lo mismo con nuestro Commodore?, ¿es que sólo sirve para perseguir a los malos o buscar tesoros? Naturalmente que no. Tu ordenador tiene muchas posibilidades, y juegos como Micro Rhythm, te las van a mostrar.

A todo aficionado a la música le hubiera gustado alguna vez tener delante, y a su disposición, una gran batería, con la que dar rienda suelta a su imaginación musical. Pues bien, ahora ese deseo puede ser realidad, gracias a Micro Rhythm, un «simulador» de una batería.

Al empezar, puede que te cueste un poco familiarizarte con las nuevas y sorprendentes posibilidades del teclado de tu ordenador, pero en breve, serás capaz de manejar perfectamente esta completísima caja de ritmos.

Puedes conseguir hasta 13 sonidos diferentes, que combinarás como tu prefieras. Bombo, tambores, pandereta, palmas, platillos, etc. están a tu disposición.

Micro Rhythm tiene tres posibilidades de utilización. En principio, y como introducción a las posibilidades del «juego», puedes escuchar una demostración de lo que se puede hacer combinando los sonidos. Cuando ya estés lo suficientemente animado, prueba tú mismo a componer tu ritmo favorito. Puedes hacerlo en tiempo real, o bien en el modo compás.

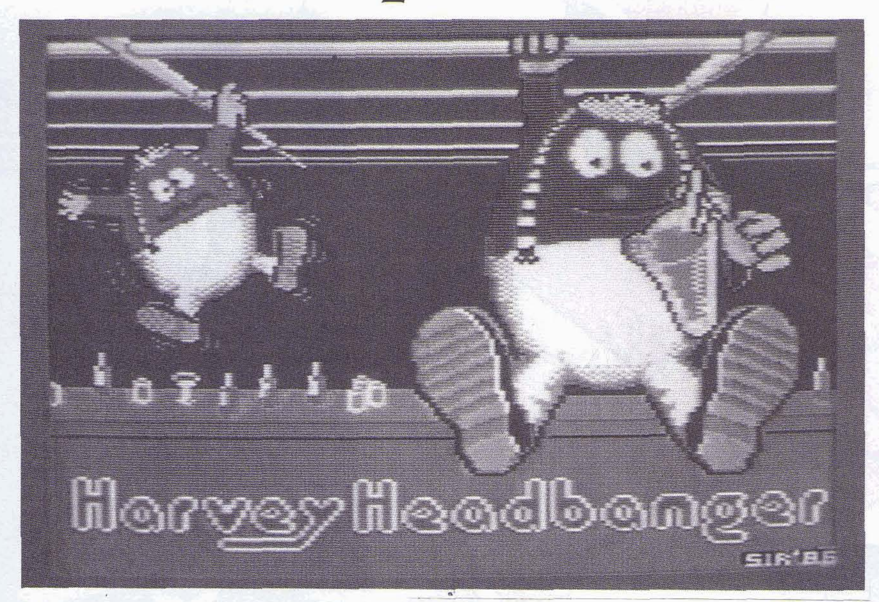
Al principio, te resultará más fácil componer en el modo compás, ya que la velocidad del ritmo está marcada, y escucharás tu melodía a su velocidad adecuada mientras la estás componiendo. Si eres más experto, utiliza el tiempo real, y luego ajusta la velocidad del ritmo.

También puedes controlar el tono, y aumentar o disminuir la duración de los sonidos, entre otras posibilidades que tú mismo irás descubriendo al jugar con Micro Rhythm.

BAR WRITE
BAR 00
SOUND Bs Bs Bs Sn Bs Bs Bs Sn Sn
FLAM
PITCH
SPEED 45 FLAM TIME 2
SONG WRITE
SONG A
07 07 25 25 07 07 25
MICRORHYTHM BY SIMON PICK (C) 1986



Harvey Headbanger



Es curioso y original este juego, donde sus dos personajes son unos gordinflones de mal genio que siempre están peleando, su única arma es «la barriga» tan grande que tienen, unida a una gran cabeza.

Siempre están en ella, tratando de empujarse para mandar al otro lo más lejos posible y así poder beberse todos lo apetecibles cócteles que aparecen de vez en cuando por la pantalla, eso sí, siempre procurando hacer el mayor ruido posible al beber.

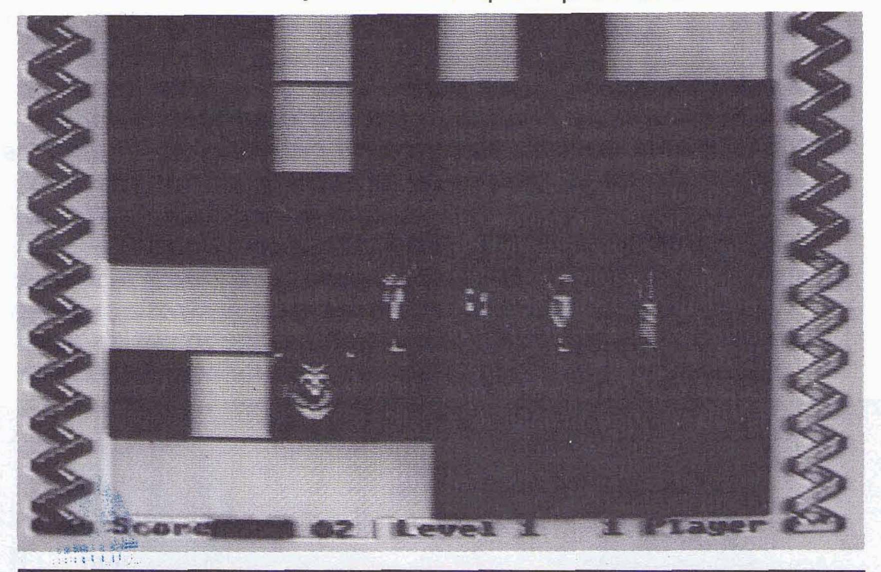
Este juego te ofrece distintas posibilidades, podrás cambiar los colores, el nivel de dificultad, quitar o poner la música de fondo, etc... Al principio te resultará un poco complicado, pero a medida que vayas jugando te irás trazando tu propia estrategia para llegar antes que tu contrario a los apetecibles cócteles, y para procurar darle en su tremenda barriga antes de que lo haga él.

Podéis participar uno o dos jugadores en esta carrera por las exóticas bebidas, o bien por medio del *joystick* o por el teclado, así que a ver quién consigue dejar K.O. a su contrario y beber mayor número de cócteles.

Cada nivel que completes te hará pasar al siguiente, que presentará mayor número de complicaciones para tus gordinflones amigos, tendrás que armarte de paciencia y practicar mucho para ser experto bebedor de cócteles y luchador de barrigas, ánimo.

Tanto los gráficos que nos presenta esta vez FIREBIRD como el sonido, están muy de acuerdo con el tipo de juego que nos proponen, es un juego rápido de un gran dinamismo que viene reflejado por los movimientos de sus pequeños personajes.

Aunque al principio te cueste llegar a controlar a tu pequeño amigo, no desesperes, ya sabes que el que la sigue la consigue, así que a practicar.



FICHA DEL JUEGO

NOMBRE:	**HARVEY HEADBANGER**
DE:	**FIREBIRD (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK, TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION							■			
GRAFICOS								■		
SONIDO								■		
GENERAL								■		



Mar
y el

Si cuando Margarita se pone el abrigo es que hace frío en el exterior, y si cuando hace frío en el exterior se consume más energía eléctrica en calefacción, el enorme desarrrollo de la ciencia en la actualidad nos permite deducir con absoluta certeza que cuando Margarita se ponga el abrigo, el consumo de energía eléctrica en la ciudad aumentará.

garita Resolutor Lógico

Así nos dice la lógica, una de las ramas de la ciencia que hace las delicias de muchas personas y que representa para otros la pura esencia de la falta de sentimientos

La anterior estrategia que hemos aplicado se basa en una intuitiva regla que recibe el pomposo nombre de **«modus ponendos ponens»** y que, con todas las demás reglas que forman la lógica aristotélica, intentan estructurar de alguna forma la manera de razonar que poseemos los humanos y que aplicamos muchas veces cada segundo. Esto lleva consigo incluso un lenguaje y unos símbolos propios y, por ejemplo, la anterior regla, desde el punto de vista formal, se puede representar como:

A --------- B	**(Si llueve, está mojado)**
A	**(Llueve)**
———	
B	**(Como conclusión, está mojado)**

¿Qué ocurre si el suelo no está mojado? (cosa que se representa por **B**, **no B**).

Pues de ahí podemos deducir que no ha llovido. Tal recibe el gracioso nombre de **«modus tollendo tollens»**, y muchas veces se usa en matemética para demostrar la falsedad de un teorema. Veamos como:

Si se cumple el teorema X, entonces tiene que cumplirse a la fuerza el teorema Y (A --------> B).

Se demuestra que el torema Y es falso (no B).

Automáticamente queda demostrado también que el teorema X ES FALSO (no A).

supongamos ahora este otro caso:

A --------> B
B

Cuya transcripción al ejemplo meteorológico sería:

Si llueve, el suelo se moja.
El suelo se moja.
¿Se deduce entonces que está lloviendo?

No. No existe regla para esto porque es falso. De hecho, es posible que esté regando la vecina de encima, o que se haya roto la presa del pantano de al lado.

La conclusión de que está lloviendo a partir de la humendad del suelo es falsa. Como hemos visto, lo único que podemos deducir es que si el suelo está seco (**no B**), entonces con absoluta suguridad, podemos decir que no ha llovido. Podemos entender ahora el pensamiento de Margarita sobre la factura de la luz que veíamos al principio.

En este ejemplo, vamos a representar el hecho de que Margarita lleve el abrigo como **A**, y el hecho de que haga frío como **B. C** representará el hecho de que el consumo de electricidad aumenta. Podemos representar la ecuación lógica como:

A --------> B

B --------> C



Vemos que Margarita se pone el abrigo. Eso significa que se da **A**, con lo que automáticamente se da **B**. Mirando en la premisa de abajo, al tener **B** deducimos **C**, que es a lo que queríamos llegar. Hemos demostrado que nuestra conclusión al principio del artículo es válida.

La fuerza de la lógica reside en que si las premisas son verdaderas, el resultado también lo será, pero, por supuesto, si partimos de hipótesis falsas, aunque las conclusiones sean correctas lógicamente hablando, no tienen porqué ser ciertas.

Tal es el caso siguiente:

Si los gansos son blancos, Julio César vendía pingüinos.

Luego Julio César vendía pingüinos.

Vamos a ver un último ejemplo que puede poner en duda la veracidad de la lógica. Todas las premisas son válidas.

Dentro de un mismo producto, los ejemplares más buscados son los más caros.

Esto en una ley general dentro del mundo económico que expresa la ley de la oferta y la demanda. (El precio de una cosa sube si hay mucha gente que quiere comprar).

Sigamos con otra premisa.

Dentro del grupo de los diamantes, los diamantes de menos de 10 pesetas son los más buscados. Esta premisa también es cierta. (A mí particularmente, me agradaría encontrar uno).

La conclusión que se desprende lógicamente de esto es que los diamantes de menos de diez pesetas son los más caros.

(Interesante conclusión que podemos probar a exponer en la joyería más cercana, con la esperanza de que el joyero crea apasionadamente en la lógica).

¿Falla la lógica? Evidentemente no. Sin embargo, la lógica aquí expuesta, la aristotélica, tiene el gran defecto de que no se ajusta al mundo real, ya que las cosas no suelen ocurrir del modo:

A --------> B

Si A, entonces B

Sino, más bien, del modo si **A**, entonces hay probabilidad de que pase **B**, pero también es posible que pasen otras cosas. Precisamente esta característica de la lógica que la aleja del mundo real, hace que se acerque al mundo del ordenador. Para este tipo de lógica, las cosas o bien son o bien no son, pero no existen cosas que se den a medias, de forma que sean en un número de casos y no sean en otro número de casos. Las que son se representan con un estado, pongamos por ejemplo un uno, y las que no son, con el otro estado, esto es, un cero.

De esta forma, el ordenador se transforma en un arma muy potente para resolver problemas de lógica. Y será esto lo que vamos a hacer esta semana. Nada menos que un «Resolutor lógico»

No obstante, debemos aclarar, para los incondicionales de la lógica, que existen otros tipos de lógica aparte de la aristotélica, como la llamada lógica difusa o lógica borrosa, donde las premisas son del tipo:

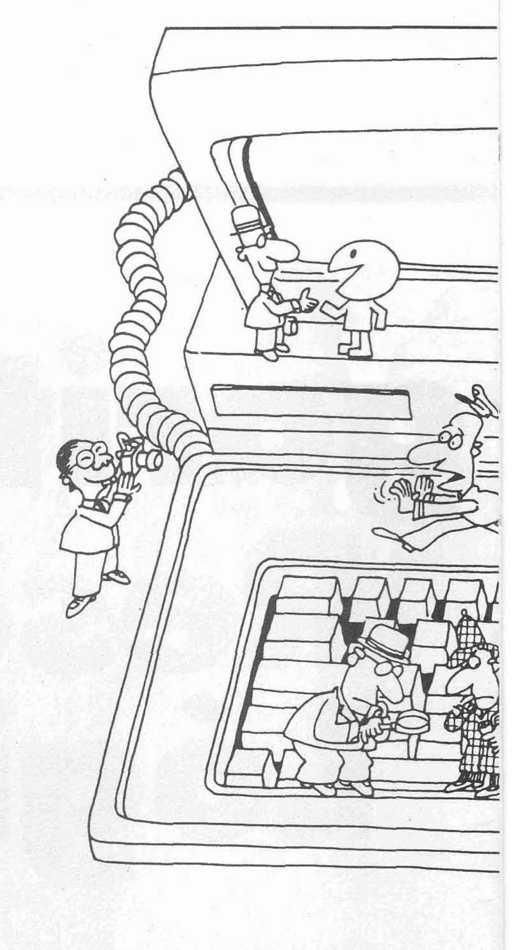
A --------> 0.83 B

que significa que si se cumple **A**, entonces **B** se cumple con una probabilidad de **0.83** (en el 83% de los casos), y **no B** se cumple en otro **17%**.

Se ha intentado acoplar al ordenador este tipo de lógica, y así muchos de los llamados «sistemas expertos» contienen reglas de este tipo. Sus deducciones lógicas son algo más complicadas, ya no son ciertas siempre, sino que es necesario reajustar las reglas de modo que nos proporcionen la probabilidad de que se cumpla la predicción (el grado de certeza).

Un claro ejemplo de esto es el **MYCIN**, unos de los primeros y más famosos sistemas expertos, para la identificación de enfermadades contagiosas, cuyas reglas eran del tipo:

«Si los análisis han dado positivo en vacilococos y tiene tos al levantarse, tiene un O.85% de probabilidad de que sea Pleurococa Tosiensis»...

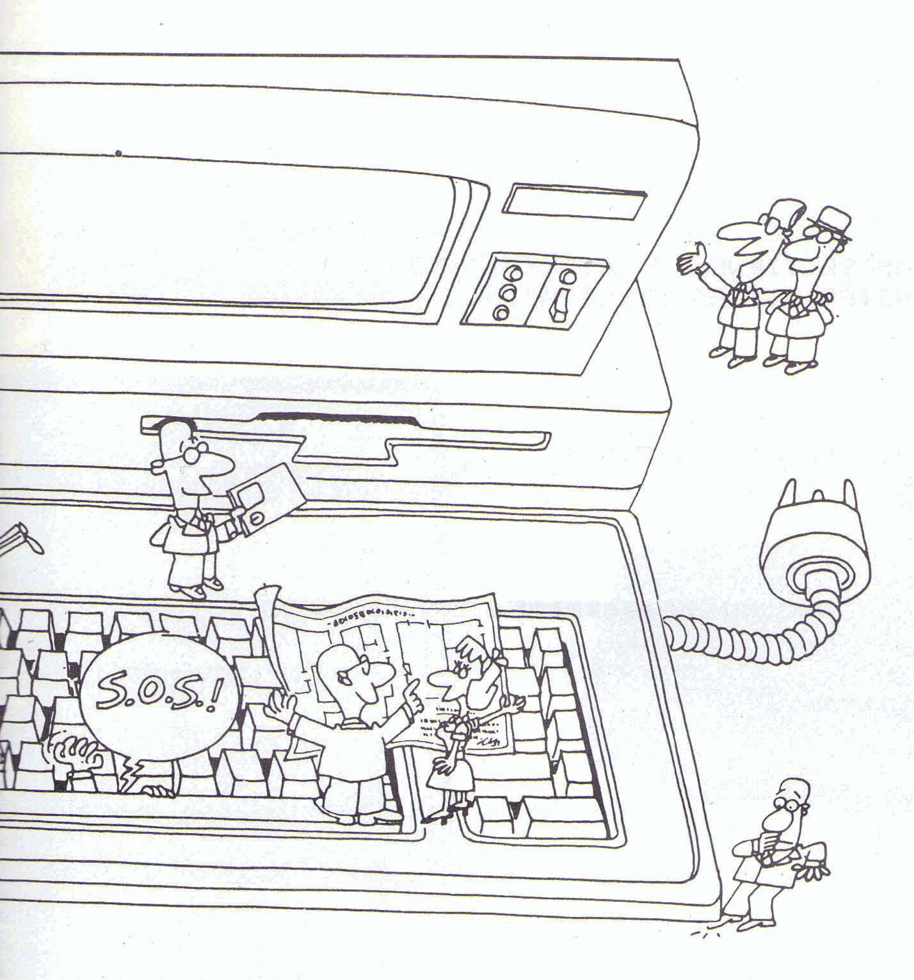


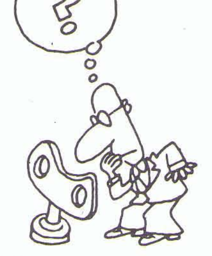
Y para identificar la enfermedad, el Mycin realiza un diálogo con el enfermo, y según las probabilidades que se vayan anotando para cada enfermedad, así será la pregunta siguiente.

No obstante, nuestra querida lógica exacta, la que Aristóteles creara en su primer tomo del «Organon», puede ser aplicable en muchos casos. Y precisamente uno de los más entretenidos consiste en resolver los pasatiempos de muchas revistas que se componen de frases variadas, y sobre las cuales nosotros debemos hacer nuestras deducciones para encontrar la solución.

Un ejemplo podría ser este.

1) **Pedro no es aviador.**
2) **Pepe no tiene el pelo rubio.**
3) **El que tiene el pelo moreno es banquero.**
4) **Rosa no es pelirroja.**
5) **El que tiene el pelo rubio no es minero**
6) **Rosa no conduce aviones.**





Que, formalizando, sería algo así:

Pd = Pedro
Ru = Rubio
Pp = Pepe
Mo = Moreno
Ro = Rosa
Pe = Pelirroja
Ba = Banquero
Mi = Minero
Av = Aviador

1) **Pd** → **no Av**
2) **Pp** → **no Ru**
3) **Mo** → **Ba**
4) **Ro** → **no Pe**
5) **Ru** → **no Mi**
6) **Ro** → **no Av**

Y las deducciones serían del tipo:

1) y 6) **Pd** ∧ **Ro** → **No Av**
(El símbolo ∧ significa en lógica ''.....y.....''Pedro y Rosa no son aviadores)
Por «tollendo tollens»,
no B → **no A**
no (**no Av**) → (**Pd** ∧ **Ro**)
Como **no** (**no Av**) es decir que la profesión no es no ser aviador eso significa entonces que si es aviador
no (**no Av**) → **Av**
Av → **no** (**Pd** ∧ **Ro**)
y como lo que dice es que ser aviador implica no ser ni Pedro ni Rosa, y nosotros sabemos que sólo hay tres personajes en este juego, entoces no ser ni Pedro ni Rosa implica ser Pepe, con lo que:
no (**Pd** ∧ **Ro**) → **Pp**
Av → **Pp**
Y de igual forma se irían calculando todas las demás respuestas.

EL PROGRAMA

Nuestro RLC (Resolutor lógico de Commodore), nos empezará pidiendo el número de nombres que hay. En este caso, habría 3 (Rosa, Pedro y Pepe), pero en un caso general no han de ser tres, ni nombres de persona, ni siquiera nombres. Representará en la matriz de resultados, cuantas filas va a haber. Por ejemplo, si lo que estamos deduciendo son las características de las provincias españolas, cada provincia será una nombre.

En el siguiente paso, se pide el número de características, sin contar el nombre, (en este caso, 2, que serían el color del pelo y el oficio), y posteriormente, el ordenador irá pidiendo todos los nombres de las personas (o cosas, o lo que sea), y después irá pidiendo los diferentes valores de las características. Por ejemplo, cuando nos pide los nombres, debemos teclear «Pedro», «Rosa» y «Pepe». (el orden no importa). En cuanto a las características, damos primero los valores de una característica, por ejemplo, el color del pelo, y tecleamos «Rubio», «Moreno» y «Pelirrojo», y después los de la siguiente característica («Aviador», «Banquero» y «Minero»). La siguiente parte serán las reglas, de los dos tipos que hemos definido, es decir, reglas positivas como «Pedro Minero», y cuando hayamos acabado de dar las positivas, el Commodores pedirá las negativas, como «Rosa no es pelirroja», que teclearemos «Rosa Pelirroja».

En este caso particular, las positivas son Moreno Banquero.
Y las negativas todas las demás.
Rubio Minero.
Pedro Aviador.
Pepe Rubio.
Rosa Pelirroja.
Rosa Aviador.
Teniendo cuidado de que las características, aviador o minero, estén tecleadas caracter a caracter, como lo fueron antes (es decir, si se teclea Minero y después minera, el ordenador no lo reconocerá).

En ambos casos, el fin de la entrada se obtendrá poniendo la palabra «FIN».

Después de esperar un periodo variable, según el número de filas y columnas (nombres y características), aparecerá por pantalla la respuesta.

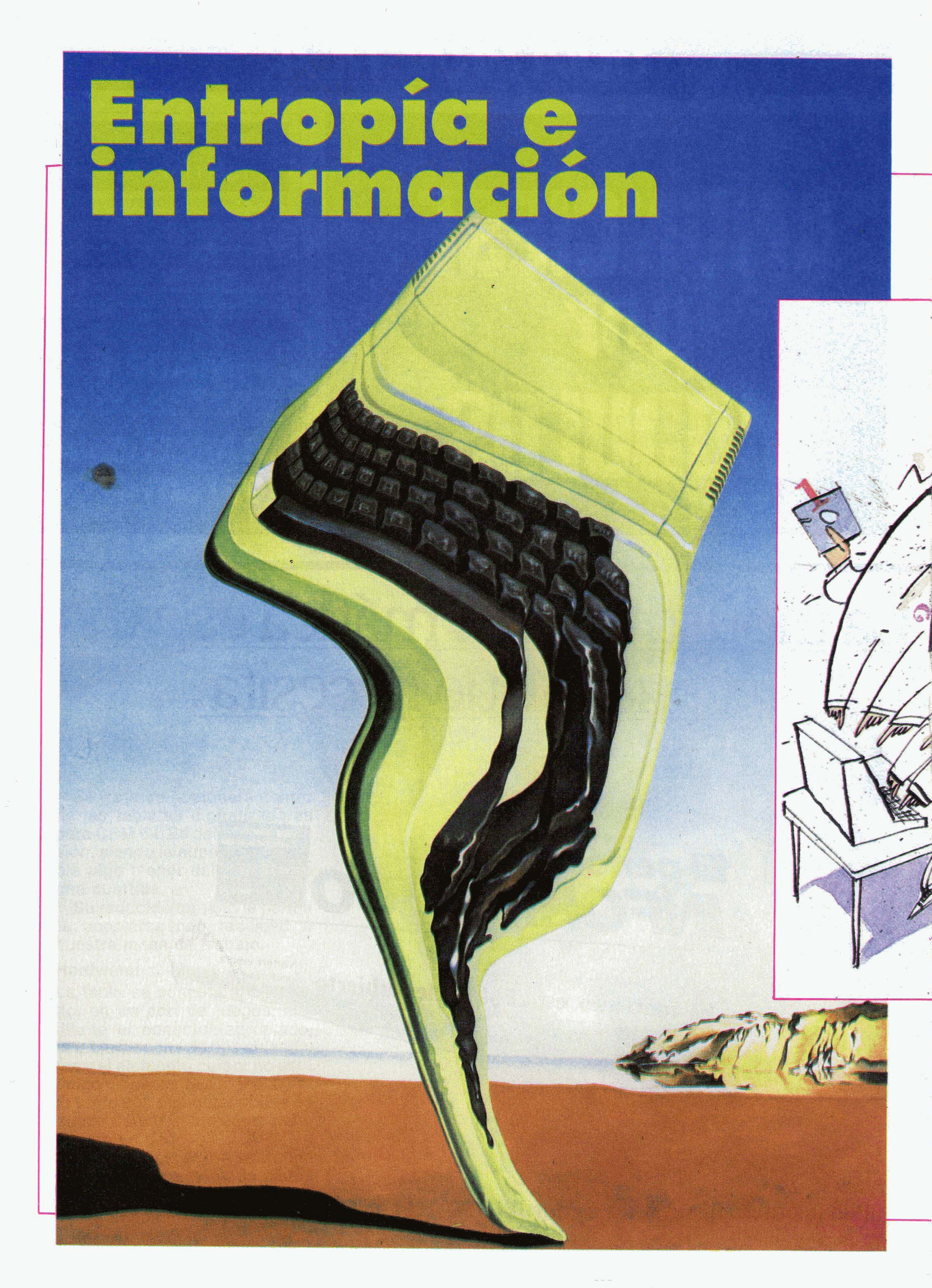
Aunque este caso es muy simple, podeis probar con ejemplos tan complicados como querais, tomando únicamente en cuenta que el tiempo aumenta exponencialmente, y así, 7 nombres y 7 características requerirán tomar las cosas con cierta dosis de paciencia. Que haya mucha suerte y mucha lógica.

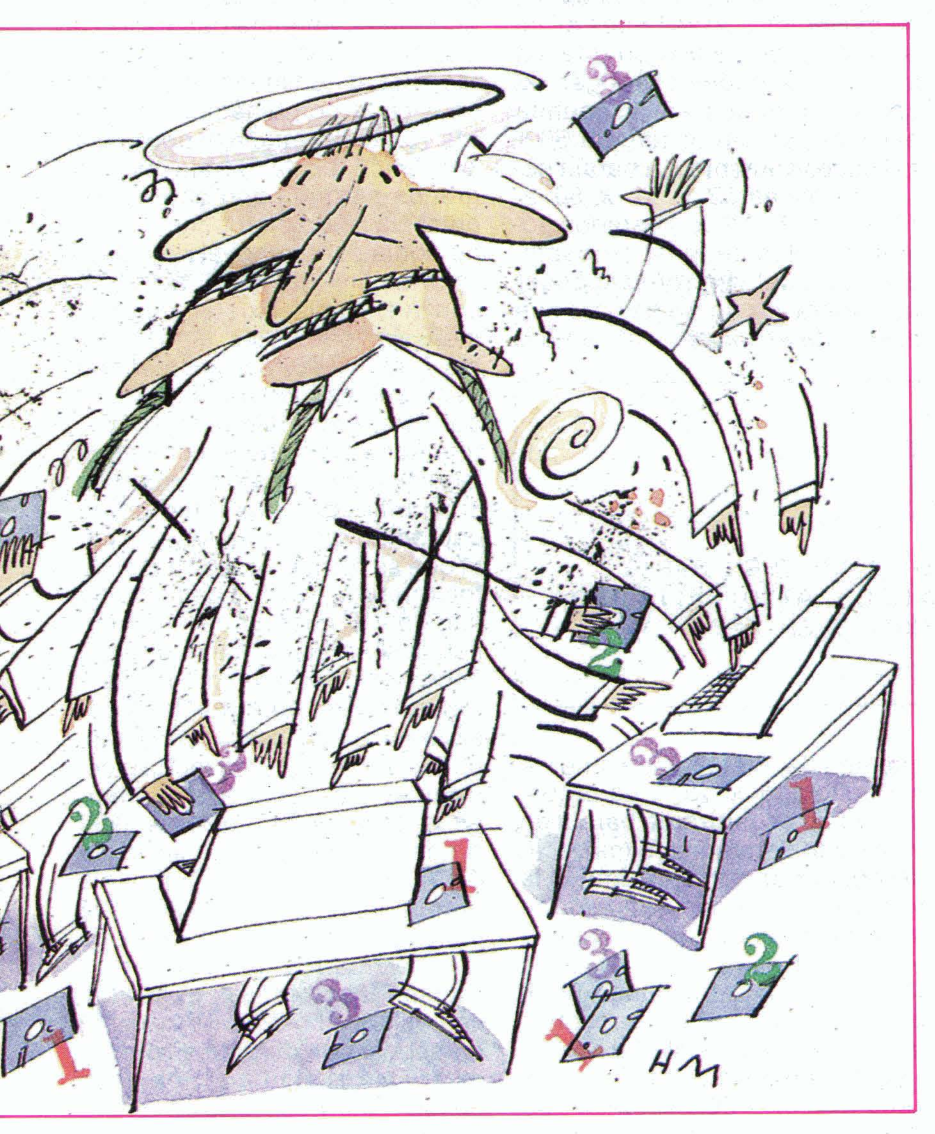
```
5 POKE 53280,6
10 INPUT "{CLR}{DOWN}CUANTOS NOMBRES TIENE TU PROBLEMA";NO
20 INPUT "{DOWN}CUANTAS CARACTERISTICAS UTILIZA";NC
30 N1=NC+1
40 DIM N$(N1,NO,10)
50 DIM A(100,N1)
60 DIM B(100,N1)
70 DIM I(N1): DIM D(100,N1): DIM E(NO,N1)
80 GOSUB 5000
90 GOSUB 6000
100 GOSUB 7000
110 GOSUB 8000
120 INDICE=1: TD=0:PRINT"{CLR}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}{RIGHT}ESTOY PENSANDO"
130 GOSUB 1000
190 GOSUB 3000
200 FOR I=1 TO NO  :PRINT"{DOWN}*";
210 FOR J=1 TO NC+1:
215 FOR Z=1 TO 10
220 PRINT N$(J,E(I,J),Z);
230 NEXT Z: NEXT J: PRINT
240 NEXT I
250 END
1000 IF INDICE>N1 THEN GOSUB 2000: RETURN
1010 I(INDICE)=1
1020 INDICE=INDICE+1: GOSUB 1000
1030 INDICE=INDICE-1
1040 I(INDICE)=I(INDICE)+1
1050 IF I(INDICE)>NO THEN RETURN
1060 GOTO 1020
2000 FOR X=1 TO TA
2010 SW=0
2020 FOR Y=1 TO N1
2025 IF A(X,Y)=0 THEN GOTO 2050
2030 IF I(Y)=A(X,Y) AND SW=0 THEN SW=1: GOTO 2020
2040 IF I(Y)<>A(X,Y) AND SW=1 THEN RETURN
2050 NEXT Y
2060 NEXT X
2200 FOR X=1 TO TB
2210 SW=0
2211 FOR Y=1 TO NC
2220 IF B(X,Y)=0 THEN GOTO 2250
2230 IF I(Y)=B(X,Y) AND SW=0 THEN SW=1: GOTO 2250
2240 IF I(Y)=B(X,Y) AND SW=1 THEN RETURN
2250 NEXT Y
2260 NEXT X
2300 TD=TD+1
2310 FOR X=1 TO N1: D(TD,X)=I(X): NEXT X
2320 RETURN
3000 FOR O=1 TO TD-NO
3005 FOR X=1 TO N1: E(1,X)=D(O,X): NEXT X: TN=1
3010 FOR X=2 TO TD
3020 FOR Y=1 TO TN
```

```
3030 FOR Z=1 TO N1
3040 IF D(X,Z)=E(Y,Z) THEN 3100
3050 NEXT Z
3060 NEXT Y
3070 TN=TN+1: FOR Y=1 TO N1: E(TN,Y)=D(X,Y): NEXT Y
3080 IF TN=NO THEN RETURN
3100 NEXT X
3110 NEXT O
3120 PRINT"■*NO TIENE SOLUCION"
3130 RETURN
5000 FOR I=1 TO NO
5010 PRINT : PRINT I;: INPUT"- NOMBRE";Y$
5020 FOR Z=1 TO LEN(Y$)
5030 N$(1,I,Z)=MID$(Y$,Z,1)
5040 NEXT Z: FOR Z=LEN(Y$)+1 TO 10:N$(1,I,Z)=" ":NEXT Z
5050 NEXT I
5100 RETURN
6000 FOR I=2 TO NC+1
6010 FOR J=1 TO NO
6020 PRINT"■CARACTERISTICA"I-1;: PRINT "TIPO"J;:INPUT Y$
6030 FOR Z=1 TO LEN(Y$)
6040 N$(I,J,Z)=MID$(Y$,Z,1)
6050 NEXT Z: FOR Z=LEN(Y$)+1 TO 10:N$(I,J,Z)=" ":NEXT Z
6060 NEXT J
6070 NEXT I
6080 RETURN
7000 PRINT"■■     AHORA PON LAS REGLAS POSITIVAS"
7010 PRINT"■ EJEMPLO:"
7020 PRINT"■ EL MORENO ES MINERO, SE ESCRIBIRIA:"
7030 PRINT"■        MORENO MINERO"
7040 PRINT"■■ PARA TERMINAR ESCRIBE FIN■■■"
7100 TA=0: DIM Y$(10): DIM O$(10)
7110 INPUT" SENTENCIA POSITIVA";Z$
7120 IF Z$="FIN" THEN RETURN
7200 I=1
7300 IF MID$(Z$,I,1)=" " THEN I=I+1: GOTO 7300
7310 J=1:I=1:FORZ=1 TO 10:Y$(Z)=" ":NEXT Z
7400 Y$(J)=MID$(Z$,I,1): J=J+1: I=I+1
7410 IF MID$(Z$,I,1)<>" " THEN 7400
7420 IF MID$(Z$,I,1)=" " THEN I=I+1: GOTO 7420
7430 J=1:FOR Z=1 TO 10:O$(Z)=" ":NEXT Z
7440 O$(J)=MID$(Z$,I,1): J=J+1: I=I+1
7450 IF MID$(Z$,I,1)<>"" THEN 7440
7500 SW=0
7505 FOR I=1 TO NC+1
7510 FOR J=1 TO NO
7520 FOR Z=1 TO 10
7530 IF Y$(Z)<>N$(I,J,Z) THEN GOTO 7600
7540 NEXT Z
7550 IF SW=0 THEN TA=TA+1:SW=1
7555 A(TA,I)=J
7600 FOR Z=1 TO 10
```

```
7610 IF O$(Z)<>N$(I,J,Z) THEN GOTO 7630
7620 NEXT Z
7622 IF SW=0 THEN TA=TA+1:SW=1
7625 A(TA,I)=J
7630 NEXT J
7640 NEXT I
7700 GOTO 7110
8000 PRINT"{CLR}{DOWN}       AHORA PON LAS REGLAS NEGATIVAS"
8010 PRINT"{DOWN} EJEMPLO:"
8020 PRINT"{DOWN} EL MORENO NO ES MINERO, SE ESCRIBIRIA:"
8030 PRINT"{DOWN}       MORENO MINERO"
8040 PRINT"{DOWN}{DOWN} PARA TERMINAR ESCRIBE FIN{DOWN}{DOWN}{DOWN}"
8100 TB=0
8110 INPUT" SENTENCIA NEGATIVA";Z$
8120 IF Z$="FIN" THEN RETURN
8200 I=1
8300 IF MID$(Z$,I,1)=" " THEN I=I+1: GOTO 8300
8310 J=1:I=1:FORZ=1 TO 10:Y$(Z)=" ":NEXT Z
8400 Y$(J)=MID$(Z$,I,1): J=J+1: I=I+1
8410 IF MID$(Z$,I,1)<>" " THEN 8400
8420 IF MID$(Z$,I,1)=" " THEN I=I+1: GOTO 8420
8430 J=1:FOR Z=1 TO 10:O$(Z)=" ":NEXT Z
8440 O$(J)=MID$(Z$,I,1): J=J+1: I=I+1
8450 IF MID$(Z$,I,1)<>"" THEN 8440
8500 SW=0
8505 FOR I=1 TO NC+1
8510 FOR J=1 TO NO
8520 FOR Z=1 TO 10
8530 IF Y$(Z)<>N$(I,J,Z) THEN GOTO 8600
8540 NEXT Z
8550 IF SW=0 THEN TB=TB+1:SW=1
8555 B(TB,I)=J
8600 FOR Z=1 TO LEN(O$)
8610 IF O$(Z)<>N$(I,J,Z) THEN GOTO 8630
8620 NEXT Z
8622 IF SW=0 THEN TB=TB+1:SW=1
8625 B(TA,I)=J
8630 NEXT J
8640 NEXT I
8700 GOTO 8110
9000 FOR I=1 TO TD
9010 FOR J=1 TO N1
9020 PRINT D(I,J);
9040 NEXT J:PRINT
9060 NEXT I
```

Entropía e información





¿Alguna vez en medio de un atasco o en una interminable cola de un supermercado os habéis preguntado lo que es el caos? Probablemente no, aunque es posible que estuvieseis inmersos en él.
Cientos de papeles revueltos sobre una mesa, buscando una camisa en un armario o sencillamente tratando de encontrar a un amigo en una fiesta, puede resultar ser un caos.

Pues si esto os vuelve a pasar otro día, podéis aprovechar para hacer un poco de filosofía. Por ejemplo, pensar acerca de la relación entre desorden e información... porque aunque en principio nos parezca que son dos conceptos lejanos el uno del otro, en realidad tienen mucho que ver. Y para entenderlo vamos a pensar un poco en lo que representa el caos. De manera cualitativa todos sabemos cuándo algo está ordenado o desordenado: en una biblioteca los libros están ordenados y en nuestra habitación en época de exámenes probablemente no lo están. Para definirlo cuantitativamente necesitamos encontrar una magnitud medible que al igual que la temperatura nos dice exactamente qué cuerpo está más caliente que otro, también nos permita comparar el grado de desorden de un sistema, ya sean libros, información, o cualquier cosa con más de un estado. Esto significa que si sólo posee un estado, un único libro, por ejemplo, no tiene sentido que nos preguntemos si está ordenado o no. Los físicos han ido descubriendo caminos para medir el caos de alguna forma: la magnitud que nos lo dice se llama entropía.

La misteriosa entropía aumenta si aumenta el desorden de un sistema, y además, si éste está aislado, la entropía no puede nunca disminuir (esto es lo que se llama la segunda ley de la termodinámica). Todo tiende al estado de máximo desorden espontáneamente: las moléculas más calientes del té, que son las que tienen más energía, se eva-

poran, mezclándose con las del aire, y al final las dos están a la misma temperatura, pero el desorden del sistema total (té más aire) ha aumentado. Llevado a un extremo, este principio nos dice que el universo llegará a su «muerte térmica», una situación en que ya no habrá movimiento ni cambio, pues todas las moléculas del universo estarán mezcladas, en el estado de mayor desorden posible. Pero bueno, aún faltan millones y millones de años...

Muy bien pero... ¿Y cómo medimos la famosa entropía? Hay varias fórmulas, pero la mejor se basa en la probabilidad de encontrar el sistema en cada una de sus posibles configuraciones. Si nosotros estamos jugando al poker, un sistema lo forman las cartas con que estoy jugando. La probabilidad de que tenga pareja (una configuración del sistema) es mucho mayor que la de que tenga repoker (otra configuración). Si ***Pi*** es la probabilidad de encontrar el sistema en el estado ***i***, entropía es la suma de ***—Pi x log(Pi)*** para todos los estados ***(log x*** es una función que nos da el Basic). Pero ¿Qué significa? Veamos algunos ejemplos. Imaginemos un sistema perfectamente ordenado, donde cada cosa tiene un único lugar, y por lo tanto sólo puede estar en ese estado, como una baraja recién sacada de un paquete nuevo. Sólo hay una configuración posible y la probabilidad de encontrar la baraja en ese estado es ***1.*** (La probabilidad va desde ***0,*** en el que el sistema no puede estar nunca, hasta ***1,*** en el que el sistema sólo está en ese estado). Como ***log (1) = 0*** tenemos que la entropía de una baraja sin estrenar es ***0.*** Pero ¿y después de mover las cartas? Pues si vamos mirando de una en una, en la primera podemos coger 52 cartas diferentes, en la segunda elección, 51, en la tercera 50 y así sucesivamente. La baraja posee entonces 52 x 51 x 50 x 49 x ... x 3 x 2 x 1 estados posibles, todos de igual probabilidad (es igual de probable coger la primera vez un AS de oros que el seis de espadas), y el número que nos da, no es precisamente pequeño ...¡***8 x*** 10^{67}! La probabilidad de encontrar la baraja en un estado determinado (es decir, de encontrar por ejemplo en la primera carta el dos de copas, en la segunda el cuatro de bastos, etc...) es de 1/ (8 x 10^{67}) (casos favorables ***1*** entre casos posibles ***8 x*** 10^{67}). Para encontrar la entropía de un sistema de cartas deberemos sumar todos esos términos de ***—1/52! x log (1/52!)*** (donde ***52!*** significa ***52 x 51 x 50 x 49...***) y lo que nos da es un número realmente grande, del orden de ***67 x*** 10^{3}.

Muy bien todo esto. Ya sabemos calcular la entropía de cualquier sistema. Pero... ¿Y qué relación existe entre información y entropía? Pues ahora vamos a ello. Ya dijimos al principio que la relación era muy directa: a mayor entropía de un sistema, más información puede transmitir. Pensemos en una moneda: puedes usar sus estados (cara o cruz) para comunicar un *bit* de datos, cero o uno. Eso es todo. Sin embargo, con una baraja puedes transmitir ***52!*** mensajes diferentes. Imagina que tuvieras un libro de códigos donde pudieras buscar qué mensaje pertenece a cada combinación de cartas. La información potencial contenida en la baraja es tan grande porque tiene una mayor entropía. Pero claro, como todo no podía ser bueno, el riesgo de «confusión» también aumenta con la entropía. Existen muchos mensajes diferentes, algunos de los cuales se diferencian en tan poco que es fácil confundir uno por otro, especialmente si existe alguna clase de «ruido» en la lectura del mensaje (el ruido es cualquier cosa que interfiera la captación de un mensaje, como el que se filtra en nuestra emisora preferida y no nos deja oír una canción, o cuando se superponen dos emisoras). Sin embargo, podemos no asignar mensajes a todos los estados posibles del sistema, y así usar algunos símbolos para detectar o incluso corregir errores. El caso más claro en la detección de errores es el *bit* de paridad. En una señal de ***8 bits*** sólo utilizamos ***7*** que nos permiten trasmitir hasta ***128*** mensajes distintos ***(***2^{7}***)*** que es la mitad de los que podríamos transmitir con esos ***8,*** y ¿qué hacemos con el octavo? Pues este *bit* será el que detecte errores, y estará a 1 si el número de unos de los restantes ***7 bits*** es impar, y a cero si ese número es par. Si en la transmisión se ha producido algún error, el número de unos no será correcto, y así si mandamos el ***1010100*** al que le corresponde un *bit* de paridad de ***1*** y en el camino se pierde un uno, lo que llegaría sería por ejemplo ***1010000*** con un *bit* de paridad de uno, que no cuadra con las reglas establecidas. El que recibe el mensaje pedirá en-

```
10 REM EXTRAPOLACION 2/3
100 DIM A(255),B(255),C(255)
200 PRINT CHR$(P)
210 GET X$: IF X$="" GOTO 210
220 PRINT CHR$(8);X$;
300 X1=ASC(X$): N=N+1: IF P=X1 THEN R=R+1
400 C(XO)=B(XO): B(XO)=A(XO): A(XO)=X1: XO=X1
410 P=A(XO): IF B(XO)=C(XO) THEN P=B(XO)
500 GOTO 200
```

tonces que se vuelva a mandar ya que la transmisión no ha sido correcta. A veces no sólo se usa un solo *bit* de paridad, sino varios, e incluso si con cada *7 bits* de información mandamos otros tres de paridad podemos saber no sólo si hay error o no, sino incluso el lugar en el que se ha producido. Esto se usa cuando es muy importante que no se produzca ningún error. Pensar lo que sudecería si fuésemos a sacar dinero de un cajero y pidiendo 4000 pts. de nuestra modesta cuenta el resultado es de un millón.

```
10 REM EXTRAPOLACION 1
100 DIM A(255)
200 PRINT CHR$(P)
210 GET X$: IF X$="" GOTO 210
220 PRINT CHR$(8);X$;
300 X1=ASC(X$): N=N+1: IF P=X1 THEN R=R+1
400 A(XO)=X1: XO=X1: P=A(XO)
500 GOTO 200
```

La entropía también es importante en la transmisión de mensajes secretos. Normalmente, para mandar estos tipos de mensajes se usan unos códigos especiales en los que el número de estados sea el más alto posible, y con la forma más complicada posible. Y, sin embargo, muchas veces los mensajes se descubren ¿Por qué? Pues por el hecho de que en cualquier idioma las letras no se utilizan con la misma frecuencia unas y otras: hay grupos de ellas que se utilizan mucho más, y así por ejemplo, la «a» está presente en nuestra lengua mucho más que la «u», así que por estadística, y probando a identificar los signos que más aparecen en el «idioma clave» las letras que más utilizamos... es muy posible que al final lo entendamos a la perfección.

Así, por ejemplo, en nuestro lenguaje hay 29 letras, en princi-

```
10 REM EXTRAPOLACION 123
100 A$="                    "
110 A$=A$+A$: A$=A$+A$: A$=A$+A$: A$=A$+A$
200 PRINT P$
210 GET X$: IF X$="" GOTO 210
220 PRINT CHR$(8);X$;
300 N=N+1: IF P$=X$ THEN R=R+1
400 A$=RIGHT$(A$,239)+X$: P=240
500 M=0:FOR I=239TO3 STEP-1:IF MID$(A$,I,1)=MID$(A$,240,1)THEN P=I+1:M=1:F=I:I=3
510 NEXT I: IF M=0 THEN 900
520 M=0:FOR I=F-1TO2 STEP-1:IF MID$(A$,I,2)=MID$(A$,239,2)THEN P=I+2:M=1:F=I:I=2
530 NEXT I: IF M=0 THEN 900
540 FOR I=F-1TO1 STEP-1:IF MID$(A$,I,3)=MID$(A$,238,3)THEN P=I+3:I=1
550 NEXT I
900 P$=MID$(A$,P,1):GOTO 200
```

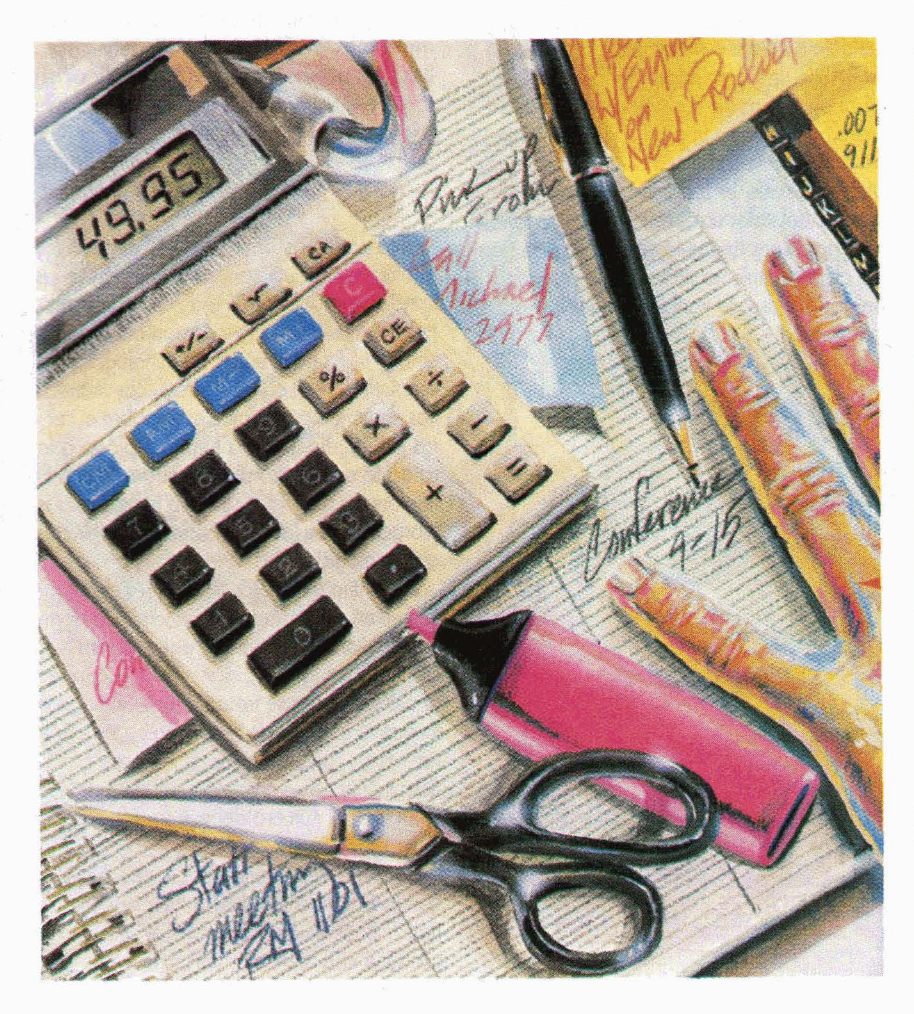
pio con la misma probabilidad de ser usadas.

La entropía para mensajes de una letra sería ***(1/29! × log (1/29!) × 29*** y para mensajes de ***N*** letras, la entropía sería ***29ⁿ = cantidad de estados;*** (1/29ⁿ × log (1/29ⁿ) = N × log 29.

Pero en la realidad, no todas son igualmente probables. Así, si tomamos una tabla de la frecuencia con que aparecen las letras en nuestro idioma, y a partir de ella computamos la suma ***P × log P,*** encontramos un nú-

mero bastante menor que el anterior, del orden de un ***30 %*** menor, aunque el valor exacto depende del tipo de texto escogido para realizar la estadística. Esto quiere decir que al haber letras que apenas se emplean y otras que se emplean mucho, globalmente el sistema «lenguaje castellano» está menos desordenado de lo que en un principio pudiera estar. Y aún más: No sólo unas letras aparecen más que otras, sino que incluso podemos predecir con mucha posibilidad de acierto qué letra seguira a otra, con lo que la entropía disminuye aún más. Por ejemplo, más de un ***50%*** de las veces en que aparece seguida de otra, ésta será una ***«a»***, (pensad por ejemplo que detrás de la letra ***«q»*** la siguiente letra es una ***«u»***).

A partir de estas ideas, se ha inventado un programa que analiza textos, y luego, mientras una palabra está siendo escrita, la máquina intenta predecir la letra siguiente antes de que el dedo humano pulse la tecla. El primer programa, el más simple, claro, escrito para ello, fue en 1978. Se llamó textrapolación, porque extrapola textos. Lo único que hace este programa es «guardar» lo que ha sucedido la última vez cuando se ha dado una letra y predice que la misma cosa volverá a suceder. Así, si nosotros tecleamos ***«la mes»***, la máquina predice que nuestra siguiente letra será una ***«a»***, porque la vez anterior escribimos ***«la mesa»***. La historia de lo que ha sido escrito está guardada en un gran vector, ***A (I)***.



Este primer programa es bastante corto y sencillo: al principio no se conoce nada y siempre se equivoca hasta que va adquiriendo la experiencia. Cuando se prueba, suele sorprender el número de veces que acierta, y podéis probarlo vosotros mismos (figura 1). El siguiente programa es algo más sofisticado y guarda en su interior no sólo el último carácter sino los últimos tres caracteres que sucedieron a la letra que se ha tecleado, y se basa en el principio de «sucederá con más probabilidad lo que ha sucedido dos de las últimas tres veces», y así, si las últimas tres veces que pusimos una ***«a»***, ésta estaba seguida de ***«o»*** una vez, y de ***«l»*** otras dos veces, lo más lógico es que la siguiente sea otra ***«l»***. El programa tarda más en aprender, pero a la larga el número de aciertos es mucho máyor (de un ***15%*** en el primero a más de un ***20%*** en el segundo). ¿Y cuál sería el programa más perfecto de todos? Pues aquel que podríamos llamar el ***«textrapolación 29»***, es decir, para cada letra guardábamos un vector de ***29*** componentes, y cada uno con el número de veces en que la siguiente letra ha sido una de las ***29*** letras. Por ejemplo, para la ***«p»***, guardaríamos en el vector, en su primera componente, el número de veces que ha aparecido la ***«a»***, en el segundo, la ***«b»***, etc., de forma que al teclear la ***«p»***, el programa «comprobaría cuál es el mayor valor de ese vector, y colocaría la letra correspondiente al índice de ese valor. Desgraciadamente, el programa en ese caso sería muy lento, aunque si se hace en ensamblador podría conseguirse casi casi un programa que acertara exactamente lo que voy a escribir después, y hasta podría acabar este texto él solito...



cartas

Números con coma flotante

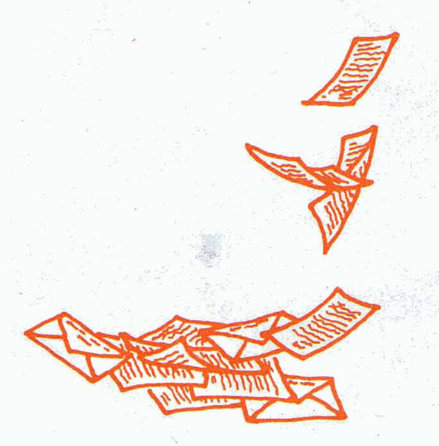
P: ¿Por qué se almacenan números con coma flotante?. Tengo entendido que se requieren 5 bytes (40 bits).Si a cada combinación de esos 40 bits le corresponde un número, y a cada número le corresponde una combinación de 40 bits, no hay ahorro de memoria (ni se amplía el intervalo de números representables) respecto al´almacenamiento en binario. Sólo hay 240 combinaciones posibles y por tanto sólo se pueden almacenar 240 números, sea en coma flotante, sea en binario. A no ser que haya combinaciones de bits que correspondan a varios números a la vez, en cuyo caso, ¿cómo los distingue el ordenador?.

Puesto que no se amplía el intervalo de números representables, intuyo que debe reportar alguna ventaja para el ordenador a la hora de operar, en cuyo caso, ¿cuál es esa ventaja?, ¿vale la pena el trabajo de conversión de decimal a coma flotante y viceversa?.

Victor M. Jiménez
Valencia

R: El almacenar números en coma flotante es necesario siempre que queramos trabajar en nuestro ordenador con números reales. Mientras no nos salgamos de ámbito de los enteros sin exponente resulta fácil el almacenar números en forma binaria valiéndonos de tan sólo uno o dos bytes, pero ¿cómo almacenar un número como 13545.23 en forma binaria?. ¿y algo como 235000000000 si disponemos de un número fijo de bytes? Para esto se inventó la coma flotante, que no es sino el equivalente binario de la notación exponencial decimal empleada en la mayoría de las calculadoras. De esta forma, utilizando sólo cinco bytes por número, podemos manejar con relativa comodidad cifras decimales dentro de unos márgenes bastante aceptables.

Sin embargo, el ordenador se enlentece mucho cuando tiene que operar con estos números en coma flotante. Es por ello por lo que muchas versiones del lenguaje BASIC poseen dos tipos de varialbes numéricas: enteras, para los número enteros, y reales para los que tienen una parte fraccionaria; el tipo de variable a emplear en cada ocasión se deja a elección del propio programador. El BASIC de Commodore también está provisto de ambos modos de almacenamiento, pero con un único tipo de varible numérica. La forma de tratar los valores en cada ocasión lo decide el propio operativo dependiendo del valor en cuestión. De esta forma el progrador obtiene lo mejor de cada tipo de almacenamiento sin tener que preocuparse por la clase de varible que debe usar en cada caso.



Señales analógicas

P: Les envío esta carta para ver si podéis explicarme lo que es una señal analógica. Esta palabra la he leído muchas veces pero nunca he comprendido lo que era aunque mis amigos me han dicho que analógico es lo contrario que digital. Y digital sí sé lo que es. Digital es la forma que tiene el ordenador para transmitir mensajes en forma de ceros y unos. Les agradecería que me explicaran lo que es analógico, pues nadie me lo ha sabido explicar.

Juan José Pérez
Segovia

R: Podríamos decir que los dispositivos analógicos se utilizan para supervisar las condiciones del mundo real: sonido, temperatura, movimiento, etc. Estas condiciones son transformadas en señales análogas a las encontradas en el mundo real. Las manecillas de un reloj analógico giran sobre el centro de un reloj de una forma análoga a la rotación de nuestro planeta sobre su eje. El teléfono es analógico, ya que transforma las vibraciones de la voz en vibraciones eléctricas analógicas.

El término analógico implica señales contínuas, en contraste al término digital que implica señales discretas, es decir, por saltos. La señal de la Televisión convencional es analógica, pues se trata de algo análogo a lo que la cámara de televisión captó originalmente, con un número de matices cromáticos infinito. Si la cámara o el Televisor fuese digital, el número de matices cromáticos sería finito.

El mundo de los ordenadores es digital y casi todo tiende a la digitalización de las señales, pues su control o procesamiento se produce con menores márgenes de error.